

# **Computational Linguistics and Automatic Reasoning**

---

# Research in Computing Science

---

## Series Editorial Board

### Editors-in-Chief:

*Grigori Sidorov (Mexico)*  
*Gerhard Ritter (USA)*  
*Jean Serra (France)*  
*Ulises Cortés (Spain)*

### Associate Editors:

*Jesús Angulo (France)*  
*Jihad El-Sana (Israel)*  
*Alexander Gelbukh (Mexico)*  
*Ioannis Kakadiaris (USA)*  
*Petros Maragos (Greece)*  
*Julian Padget (UK)*  
*Mateo Valero (Spain)*

### Editorial Coordination:

*Alejandra Ramos Porras*  
*Carlos Vizcaino Sahagún*

*Research in Computing Science* es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 147, No. 6**, junio de 2018. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

**Editor responsable:** *Grigori Sidorov, RFC SIGR651028L69*

**Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 147, No. 6**, June 2018. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

# Computational Linguistics and Automatic Reasoning

Grigori Sidorov (ed.)



Instituto Politécnico Nacional  
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación  
México 2018

**ISSN: 1870-4069**

---

Copyright © Instituto Politécnico Nacional 2018

Instituto Politécnico Nacional (IPN)  
Centro de Investigación en Computación (CIC)  
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal  
Unidad Profesional “Adolfo López Mateos”, Zacatenco  
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico



## Editorial

This volume of the journal “Research in Computing Science” contains selected papers related to computational linguistics, automatic reasoning and their applications. The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee or additional reviewers. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers. It is worth noting that various papers for this volume were rejected.

The volume contains 25 papers, which treat various aspect of modelling using Artificial Intelligence techniques in computational linguistics and automatic reasoning. Among the topics of the volume, there are semantic similarity, natural language interfaces, deception detection, sentiment analysis, opinion mining, dialogue management, detection of author profiles, etc. Several papers deal with fuzzy logic and its applications.

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial) and COMIA for their support in preparation of this volume.

The entire submission, reviewing, and selection process, as well as preparation of the proceedings, were supported for free by the EasyChair system ([www.easychair.org](http://www.easychair.org)).

*Grigori Sidorov*  
Instituto Politécnico Nacional  
México  
Guest Editor

June 2018



## Table of Contents

	Page
Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions .....	11
<i>Andrés Dobó</i>	
Towards Topic Detection in Plain Documents Using Key Terminology Extraction Supported by Institute of Technology Tallaght.....	27
<i>Juan Huetle Figueroa, Fernando Perez Tellez, David Pinto</i>	
El método de anagramas: un rápido y novedoso algoritmo para generar jugadas de scrabble .....	41
<i>Alejandro González Romero, René Alquézar Mancho, Arturo Ramírez Flores, Francisco González Acuña, Ian García Olmedo</i>	
Identificación del perfil de usuario en Twitter utilizando recursos semánticos .....	57
<i>J. Víctor Carrera Trejo, Miguel Á. Álvarez Carmona, Luis Villaseñor Pineda</i>	
Identificación de relaciones taxonómicas de dominio usando métricas textuales .....	71
<i>Yuridiana Alemán, María Somodevilla, Darnes Vilariño</i>	
Sistema para la generación personalizada de resúmenes a partir de múltiples documentos .....	85
<i>Orlando Hernández Hernández, Esaú Villatoro Tello, Christian Lemaitre León</i>	
Identificación de etiquetas semánticas para su uso en diálogos.....	99
<i>Andrés Vázquez, David Pinto, Darnes Vilariño</i>	
Representaciones vectoriales de palabras de un corpus de normas de asociación .....	109
<i>Jorge Reyes Magaña, Helena Gómez Adorno, Gemma Bel Enguix, Gerardo Sierra</i>	
Medidas de similitud semántica aplicadas a una ontología de dominio .....	119
<i>Aimee Cecilia Hernández García, Mireya Tovar Vidal, José de Jesús Lavalle Martínez</i>	

Detección automática de engaño en notas de opinión a partir de técnicas de perfilado de autores.....	133
<i>Jonathan Serrano-Pérez, Javier Sánchez Junquera, Hugo Jair Escalante Balderas, Luis Villaseñor Pineda</i>	
Evaluación de candidatos para la retroalimentación de corpus por medio de bagging .....	145
<i>Cecilia Reyes Peña, David Pinto Avendaño, Darnes Vilariño Ayala</i>	
Interfaz de lenguaje natural para consultar cubos multidimensionales utilizando procesamiento analítico en línea .....	153
<i>J. A. Porras Medrano, R. Florencia Juárez, G. Rivera Zárate, V. García Jiménez</i>	
Minería de opiniones aplicada a la evaluación docente de la UACJ .....	167
<i>Rafael Jiménez Castro, Vicente García, Rogelio Florencia Juárez, Gilberto Rivera Zarate, Francisco López Orozco</i>	
Uso de analizador de emociones en sistemas educativos inteligentes .....	179
<i>María Lucía Barrón Estrada, Ramón Zatarain Cabada, Sandra Lucía Ramírez Ávila, Raúl Oramas-Bustillos, Mario Graff Guerrero</i>	
Interfaz de lenguaje natural para deducción de información almacenada en ontologías .....	191
<i>Alejandro Solís Sánchez, Rogelio Florencia Juárez, Juan Carlos Acosta Guadarrama, Francisco López Orozco</i>	
Una representación basada en esquemas preconceptuales de eventos determinísticos y aleatorios tipo señal desde dominios de software científico .....	209
<i>Paola Andrea Noreña Cardona, Carlos Mario Zapata Jaramillo</i>	
Un método para el análisis de sentimientos bajo un enfoque supervisado usando recursos léxicos .....	223
<i>Antonio Hernández Ambrocio, Gabriela Ramírez de la Rosa, Esau Villatoro Tello</i>	
Arquitectura para el análisis de estados financieros XBRL publicados por empresas en México utilizando lógica difusa .....	237
<i>Cristian Noé Enríquez Marcial, Hilarion Miño Contreras, José Luis Sánchez Cervantes, Lisbeth Rodríguez Mazahua, Giner Alor Hernández</i>	

Desarrollo de un sistema domótico con controlador difuso y controlador manual, implementado en LabView y Arduino IDE .....	251
<i>José Alberto Vázquez Fernández, David Tinoco Varela</i>	
TFL <sup>PL</sup> : Programación con lógica de términos.....	267
<i>J. Martín Castro-Manzano, L. Ignacio Lozano-Cobos</i>	
Aplicación de lógica difusa en el proceso de shot peening del aluminio 2024-T351 .....	287
<i>Alicia Guadalupe Lazcano Herrera, Sandra Silvia Roblero Aguilar, José Solís Romero, Héctor Rafael Orozco Aguirre, Víctor Augusto Castellanos Escamilla</i>	
Identificación y control difuso en el diagnóstico de fallas para sistemas una entrada-una salida: aplicado a la dirección de un robot tractor.....	301
<i>Raúl Cortes-Gutiérrez, Julio C. Ramos Fernández, Juan David Padre Nonthe, Marco A. Márquez Vera, Filiberto Muñoz Palacios</i>	
Regulación de voltaje de un convertidor buck-boost mediante su modelo difuso inverso .....	317
<i>Nadia S. Zúñiga Peña, Marco A. Márquez Vera, Julio C. Ramos Fernández, Luis F. Cerecero Natale, Filiberto Muñoz Palacios</i>	
Representación de eventos de ruido ambiental a partir de esquemas preconceptuales y buenas prácticas de educación geoespacial de requisitos.....	329
<i>Claudia Elena Durango Vanegas, Paola Andrea Noreña Cardona, Carlos Mario Zapata Jaramillo</i>	
Diseño de un sistema de suministro de energía para vehículos eléctricos usando lógica difusa .....	345
<i>Ismael Osuna Galán, Yolanda Pérez Pimente, Juan Villegas Cortez, Carlos Avilés Cruz</i>	



# Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions

András Dobó

University of Szeged, Institute of Informatics, Szeged,  
Hungary

dobo@inf.u-szeged.hu

**Abstract.** Smoothing is an essential tool in many NLP tasks, therefore numerous techniques have been developed for this purpose in the past. One of the most widely used smoothing methods are the Kneser-Ney smoothing (KNS) and its variants, including the Modified Kneser-Ney smoothing (MKNS), which are widely considered to be among the best smoothing methods available. Although when creating the original KNS the intention of the authors was to develop such a smoothing method that preserves the marginal distributions of the original model, this property was not maintained when developing the MKNS. In this article I would like to overcome this and propose such a refined version of the MKNS that preserves these marginal distributions while keeping the advantages of both previous versions. Beside its advantageous properties, this novel smoothing method is shown to achieve about the same results as the MKNS in a standard language modeling task.

**Keywords:** multi-D Kneser-Ney smoothing, original marginal distributions.

## 1 Introduction

The goal of smoothing is to overcome data sparsity, which poses a huge problem in numerous tasks, including a vast number of NLP problems. A very good example of this is language modeling, where the task is to learn the probability of word sequences given some training data: using lower order models for this purpose do not provide sufficient context, while choosing large models will usually suffer from insufficient training data (for an  $n$ -gram model there are  $|v|^n$  distinct  $n$ -gram types, where  $|v|$  is the size of the vocabulary). Due to this, most of the values in a basic  $n$ -gram model are equal to zero, which produces zero probabilities for most word sequences when simply using maximum-likelihood estimation. To overcome this, smoothing techniques have been widely used since decades, decreasing the probability of seen events and redistributing the gained probability mass among unseen events so as to avoid zero probabilities during prediction.

Due to their importance, smoothing methods have received considerable attention in the past. One of the most widely used group of smoothing methods are

of the type absolute discounting [10], that are simple but still very powerful and efficient methods. The Kneser-Ney smoothing (KNS) [8], and its multi-discount variant, the Modified Kneser-Ney smoothing (MKNS) [1] are widely considered to be one of the best smoothing algorithms since a long time [1,6,14,12,13,16].

Although the probability of atomic events changes during smoothing as a necessary consequence, the marginal probabilities do not necessarily need to change, where the marginal probabilities are the probabilities obtained by summing out the probabilities of an event with respect to other events:

$$P(Y) = \sum_{z \in Z} P(Y, z). \quad (1)$$

One of the key motivations when developing the KNS was that it should preserve the marginal distributions of the original model, meaning that the obtained model satisfies the following equation:

$$\frac{c(w_i)}{\sum_{w_i} c(w_i)} = \sum_{w_{i-1}} p(w_i | w_{i-1}) p(w_{i-1}). \quad (2)$$

This is very advantageous in many cases, and under certain assumptions, an optimal model can only be obtained by satisfying this property, as discussed by Goodman in the extended version of his paper [6]. Hence Goodman comes to the conclusion that under these assumptions any smoothing method not preserving the original marginals can be improved by modifying it to preserve them. Despite this fact, many frequently used smoothing techniques, including the MKNS, do not satisfy this property: when Chen and Goodman [1] refined the original KNS by introducing three discount parameters instead of just one, they did not adjust the lower-order distributions according to this change, which resulted in the loss of the original marginals in the smoothed model.

Within this article I present such a novel smoothing method based on the MKNS, that keeps all the advantages of both the KNS and the MKNS, while also preserving the original marginal distributions. Section 2 gives a general overview of smoothing methods and introduces the problem of preserving the marginal distributions in detail. The presentation of my novel method (called MDKNS-POMD) follows in Section 3, which is evaluated on a standard language modeling task in Section 4. Section 5 gives a short summary and draws conclusions.

## 2 Background

One of the earliest and simplest smoothing algorithms is called Add-k smoothing [9,1]. This basically adds a fixed constant value (often simply 1) to the count of each observed and unobserved event, and computes the probabilities on these modified counts. As in case of this model too, in the rest of the article I will present each formula and example adapted to bigram language modeling (when not noted otherwise). However, all methods can be used generally on higher-order models and on other applicable data types too. The bigram formula for the



Add-k smoothing, with a smoothing parameter  $\delta$  and a vocabulary size of  $|V|$ , is as follows:

$$p_{Add}(w_i|w_{i-1}) = \frac{\delta + c(w_{i-1}w_i)}{\delta|V| + \sum_{w_i} c(w_{i-1}w_i)}. \quad (3)$$

Due to its simplicity, this method is used very widely and can actually work very well in some cases, especially if there are not too many zero counts. However, in case of most problems, especially with huge models full of zeros (such as language modeling), it overweighs unseen events and does not work too well. Besides, it also does not keep the original marginals.

Another frequently used method is the Good-Turing smoothing (GT) [5], trying to estimate the probability of words that occurred  $r$  times using the frequency of words seen  $(r+1)$  times:

$$r^* = (r + 1) \frac{n_{r+1}}{n_r}, \quad (4)$$

where  $n_r$  is the number of  $n$ -grams that are present in the corpora exactly  $r$  times. Based on this the probability of a bigram occurring  $r$  times is given as:

$$p_{GT}(w_i|w_{i-1}) = \frac{r^*}{\sum_{w_i} c(w_i)}. \quad (5)$$

Although this method has a very good intuition and can sometimes work quite well, it does not combine higher-order models with lower-order models, and just uses the same general smoothing for all words with count  $r$ . Therefore this is usually outperformed by more sophisticated methods.

A very popular category of smoothing methods try to estimate the probabilities in an  $n$ -gram model by also making use of information from an  $(n-1)$ -gram model. This is advantageous as there are much less  $(n-1)$ -gram types than  $n$ -gram types, so the number of zeros in the  $(n-1)$ -gram model is much less than in case of the  $n$ -gram model. These techniques either back off to the lower order model or interpolate the higher-order model with it. As interpolation is fairly consistently more successful than backing off [1,6], in the rest of the article I will only consider this version of each smoothing algorithm. Those smoothing methods that back off from higher-order models or interpolate them with lower order models can be recursively applied to the lower order models too, which further helps eliminating the zero probabilities and usually helps to achieve better results.

One of the easiest ways to create an interpolated model is by simple absolute discounting (Abs) [10]. The motivation behind this was that looking at the results of other smoothing methods, such as the Good-Turing smoothing, one can often notice that it is as if the same value was simply subtracted from the count of each seen event ( $D < 1$ ), hence it would be easier to just simply do this instead of doing more complex calculations:

$$p_{Abs}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i) - D}{\sum_{w_i} c(w_{i-1}w_i)} + (1 - \lambda_{w_{i-1}}) p_{Abs}(w_i). \quad (6)$$

Despite its simplicity, absolute discounting can work quite well, and it was the basis for many of the most successful discounting methods currently in use. Among these techniques are the Witten-Bell [15], the Jelinek-Mercer [7] and the Kneser-Ney smoothing (KNS) [8], and their variants.

The motivation behind the original KNS was to implement absolute discounting in such a way that would keep the original marginals unchanged, hence preserving all the marginals of the unsmoothed model. Their model is as follows (actually, the original article [8] only presented a backoff version, and the interpolated version shown here was only introduced by [1]):

$$p_{KNS}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i) - D}{\sum_{w_i} c(w_{i-1}w_i)} + \gamma_{KNS}(w_{i-1}) p_{KNS}(w_i). \quad (7)$$

The  $\gamma_{KNS}(w_{i-1})$  serves as normalization and should be chosen in a way so that the distributions of the words sum up to 1. For that the sum of the  $\gamma_{KNS}(w_{i-1})$  weights for a word should be the same as the sum of the discounts subtracted from the probabilities of the word:

$$\gamma_{KNS}(w_{i-1}) = \frac{N_{1+}(w_{i-1}) D}{\sum_{w_i} c(w_{i-1}w_i)}, \quad (8)$$

with the  $N$  functions defined as follows:

$$\begin{aligned} N_c(w_{i-1}) &= |\{w_i : c(w_{i-1}w_i) = c\}|, \\ N_{c+}(w_{i-1}) &= |\{w_i : c(w_{i-1}w_i) \geq c\}|, \\ N_c(.w_i) &= |\{w_{i-1} : c(w_{i-1}w_i) = c\}|, \\ N_{c+}(.w_i) &= |\{w_{i-1} : c(w_{i-1}w_i) \geq c\}|, \\ N_c(..) &= |\{w_{i-1}, w_i : c(w_{i-1}w_i) = c\}|, \\ N_{c+}(..) &= |\{w_{i-1}, w_i : c(w_{i-1}w_i) \geq c\}|. \end{aligned} \quad (9)$$

In this model the discount parameter and the lower-order distribution are the free parameters, as the count values are given by the training data and the normalization is given based on the other parameters. Therefore one can implement different versions of this model by changing these two free parameters. By choosing the right lower-order distribution it is possible to preserve the marginal probabilities. To achieve this one has to define the lower order distribution to return a probability of  $p$  for a word  $w_i$ , where  $p$  is the proportion of the discounts subtracted from the  $c(.w_i)$  counts as compared to the discounts subtracted from all the  $c(..)$  values:

$$p_{KNS}(w_i) = \frac{N_{1+}(.w_i)}{N_{1+}(..)}. \quad (10)$$

However, please note that the marginals are only preserved in case of bigram models or in case of such higher-order models where the highest order model is simply interpolated with the unsmoothed second-to-highest order model. In case the second-to-highest order model is smoothed recursively the same way, then this property of the KNS is lost.

Chen and Goodman [1] introduced an improved version of the original KNS by changing one of its free parameters, namely the discount parameter. They showed that the optimal discount values for counts of 1 and 2 are very different from the optimal discount value for higher counts. Therefore they proposed to have three discounting parameters ( $D_1 < 1$ ,  $D_2 < 2$  and  $D_{3+} < 3$ ) instead of just one, for counts of 1, 2 and at least 3, respectively:

$$p_{MKNS}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i) - D(c(w_{i-1}w_i))}{\sum_{w_i} c(w_{i-1}w_i)} + \gamma_{MKNS}(w_{i-1})p_{MKNS}(w_i), \quad (11)$$

where

$$D(x) = \begin{cases} 0 & \text{if } x = 0, \\ D_1 & \text{if } x = 1, \\ D_2 & \text{if } x = 2, \\ D_{3+} & \text{if } x \geq 3. \end{cases} \quad (12)$$

With this modification, the normalization factor, in order to have all the word distributions sum up to 1, should be defined as follows:

$$\gamma_{MKNS}(w_{i-1}) = \frac{D_1 N_1(w_{i-1}\cdot) + D_2 N_2(w_{i-1}\cdot) + D_{3+} N_{3+}(w_{i-1}\cdot)}{\sum_{w_i} c(w_{i-1}w_i)}. \quad (13)$$

As also noted earlier, the unigram distribution remains the same as it was in case of the KNS:

$$p_{MKNS}(w_i) = p_{KNS}(w_i). \quad (14)$$

The optimal discount parameters for the KNS and MKNS models can be estimated as follows [1]:

$$\begin{aligned} D &= \frac{n_1}{n_1 + 2n_2}, \\ D_1 &= 1 - 2D \frac{n_2}{n_1}, \\ D_2 &= 2 - 3D \frac{n_3}{n_2}, \\ D_{3+} &= 3 - 4D \frac{n_4}{n_3}. \end{aligned} \quad (15)$$

where  $n_r$  represents the total number of n-grams with a frequency of  $r$ .

Although the discounting method in the MKNS is different then in the KNS, the authors left the calculation of the lower-order distributions unchanged, which results in not preserving the original marginal distributions. There already exist a couple of studies discussing this issue. Some simply note this fact [14,12,13], without presenting any detailed discussion about it, while others also get into more detail.

For example, Zhang and Chiang [16] also note that this property of the MKNS can be resolved, but they do not provide a solution for this. Further, Chen and Rosenfeld [2] note that using maximum entropy (ME) techniques one can obtain such models that preserve the original marginals. However, they do not apply this to the MKNS and only discuss in detail a Fuzzy ME model that only approximately preserves them. Although Roark et al. [11] presents such a method that takes an arbitrary backoff smoothing model and transforms it into a model that preserves the original marginals, they do not test this technique on the MKNS model. So despite the earlier studies considering this problem, to my best knowledge, my study is the first to derive the solution for the MKNS and to perform thorough tests comparing the original KNS and MKNS methods with this new method. Therefore this study is novel in this respect.

To help better understand the difference between the smoothing methods and to give an easy insight into what preserving or not preserving the marginals means, I hereby present the joint and marginal counts of a sample bigram maximum likelihood model, unsmoothed and smoothed with KNS and MKNS, trained on the same small text in Tables 1, 2 and 3, respectively. In case of every table, each row corresponds to a value of  $x$ , each column represents a value of  $y$ , with the cells containing the  $c(x,y)$  values. The  $\langle s \rangle$  and  $\langle /s \rangle$  symbols are special symbols representing the beginning and end of the sentences, respectively. Inside the tables counts are presented instead of probabilities, as this way it is much easier to see whether the original marginals are preserved after smoothing or not. The sum of the counts in each column (which corresponds to the respective marginal) are presented at the end of them.

### 3 My Novel Method

As previously presented, the MKNS is widely considered to be one of the best smoothing algorithms. However, despite the original motivation for its base variant (KNS), it does not have the property of keeping the original marginals unchanged. My novel method, which is called Multi-D Kneser-Ney Smoothing Preserving the Original Marginal Distributions (MDKNSPOMD), was developed to overcome this problem.

Its basis comes from the MKNS, with the idea for maintaining the marginal distributions from the original KNS, by changing one of the free parameters of the MKNS, namely the lower-order distribution. It is easy to see that the marginal distributions in a bigram model can be preserved if the lower order distribution is designed in a way that it returns a probability of  $p$  for a word  $w_i$ , where  $p$  is the proportion of the discounts subtracted from the count of the bigrams ending with  $w_i$  as compared to all the discounts subtracted at the whole bigram level. This can easily be derived mathematically for the MKNS, in a similar manner as Chen and Goodman [1] did it for the KNS. The derivation starts with the following equation:

**Table 1.** Joint and marginal counts of a simple maximum likelihood model trained on the sample text.

c(x, y)	<s>	a	b	c	d	e	</s>	
<s>	0	2	3	5	0	1	0	11
a	0	4	1	4	3	8	1	21
b	0	7	2	1	0	0	4	14
c	0	2	5	2	0	4	2	15
d	0	1	0	0	2	0	3	6
e	0	5	3	3	1	6	1	19
</s>	0	0	0	0	0	0	0	0
	0	21	14	15	6	19	11	86

**Table 2.** Joint and marginal counts of a simple maximum likelihood model with KNS trained on the sample text.

c(x, y)	<s>	a	b	c	d	e	</s>	
<s>	0.00	1.95	2.89	4.89	0.16	0.84	0.26	11.00
a	0.00	4.11	1.03	4.03	2.87	7.95	1.03	21.00
b	0.00	6.95	1.89	0.89	0.16	0.21	3.89	14.00
c	0.00	2.03	4.96	1.96	0.20	3.89	1.96	15.00
d	0.00	0.87	0.20	0.20	1.75	0.16	2.83	6.00
e	0.00	5.11	3.03	3.03	0.87	5.95	1.03	19.00
</s>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	21.00	14.00	15.00	6.00	19.00	11.00	86.00

**Table 3.** Joint and marginal counts of a simple maximum likelihood model with MKNS trained on the sample text.

c(x, y)	<s>	a	b	c	d	e	</s>	
<s>	0.00	2.01	2.09	4.09	0.55	1.36	0.91	11.00
a	0.00	3.90	2.06	3.61	2.04	7.32	2.06	21.00
b	0.00	6.27	1.83	1.54	0.55	0.73	3.09	14.00
c	0.00	2.40	4.41	2.15	0.74	3.16	2.15	15.00
d	0.00	1.33	0.58	0.58	1.27	0.47	1.76	6.00
e	0.00	4.90	2.61	2.61	1.49	5.32	2.06	19.00
</s>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	20.80	13.58	14.58	6.63	18.36	12.04	86.00

$$\frac{c(w_i)}{\sum_{w_i} c(w_i)} = \sum_{w_{i-1}} p(w_i|w_{i-1}) p(w_{i-1}), \quad (16)$$

in which it is possible to express  $p(w_{i-1})$  by its empirical estimation from the training data:

$$p(w_{i-1}) = \frac{c(w_{i-1})}{\sum_{w_{i-1}} c(w_{i-1})} \quad (17)$$

to get (after some simplification):

$$c(w_i) = \sum_{w_{i-1}} c(w_{i-1}) p(w_i|w_{i-1}). \quad (18)$$

Then  $p(w_i|w_{i-1})$  can be substituted with its formula in MKNS (Equation 11):

$$c(w_i) = \sum_{w_{i-1}} c(w_{i-1}) \left( \frac{c(w_{i-1}w_i) - D(c(w_{i-1}w_i))}{\sum_{w_i} c(w_{i-1}w_i)} + \gamma(w_{i-1}) p(w_i) \right), \quad (19)$$

after which  $\gamma(w_{i-1})$  can also be expressed as it is for MKNS in Equation 13, and a couple of simplifying steps can be made to obtain:

$$c(w_i) = \sum_{w_{i-1}} c(w_{i-1}) \left[ \frac{c(w_{i-1}w_i) - D(c(w_{i-1}w_i))}{c(w_{i-1})} + \frac{D_1N_1(w_{i-1}.) + D_2N_2(w_{i-1}.) + D_{3+}N_{3+}(w_{i-1}.)}{c(w_{i-1})} p(w_i) \right], \quad (20)$$

$$c(w_i) = \sum_{w_{i-1}} \left[ c(w_{i-1}w_i) - D(c(w_{i-1}w_i)) + (D_1N_1(w_{i-1}.) + D_2N_2(w_{i-1}.) + D_{3+}N_{3+}(w_{i-1}.) p(w_i)) \right], \quad (21)$$

$$c(w_i) = c(w_i) - \left( \sum_{w_{i-1}} D(c(w_{i-1}w_i)) \right) + p(w_i) \sum_{w_{i-1}} (D_1N_1(w_{i-1}.) + D_2N_2(w_{i-1}.) + D_{3+}N_{3+}(w_{i-1}.) p(w_i)). \quad (22)$$

From here  $p(w_i)$  can be easily expressed as:

$$p(w_i) = \frac{\sum_{w_{i-1}} D(c(w_{i-1}w_i))}{\sum_{w_{i-1}} (D_1N_1(w_{i-1}.) + D_2N_2(w_{i-1}.) + D_{3+}N_{3+}(w_{i-1}.) p(w_i))}, \quad (23)$$

and the sums can be rewritten to get the following form:

$$p(w_i) = \frac{D_1 N_1(.w_i) + D_2 N_2(.w_i) + D_{3+} N_{3+}(.w_i)}{D_1 N_1(..) + D_2 N_2(..) + D_{3+} N_{3+}(..)}. \quad (24)$$

This proves that the MKNS smoothing model with the modification of using the above lower-order distribution will preserve the original marginal probabilities when interpolating with the above unigram distribution, and there are no other ways to achieve this. So this gives the following final form for the MDKNSPOMD for a bigram language model:

$$p_{MDKNSPOMD}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i) - D(c(w_{i-1}w_i))}{\sum_{w_i} c(w_{i-1}w_i)} + \gamma_{MDKNSPOMD}(w_{i-1}) p_{MDKNSPOMD}(w_i), \quad (25)$$

$$\gamma_{MDKNSPOMD}(w_{i-1}) = \frac{D_1 N_1(w_{i-1}.) + D_2 N_2(w_{i-1}.)}{\sum_{w_i} c(w_{i-1}w_i)} + \frac{D_{3+} N_{3+}(w_{i-1}.)}{\sum_{w_i} c(w_{i-1}w_i)}, \quad (26)$$

$$p_{MDKNSPOMD}(w_i) = \frac{D_1 N_1(.w_i) + D_2 N_2(.w_i) + D_{3+} N_{3+}(.w_i)}{D_1 N_1(..) + D_2 N_2(..) + D_{3+} N_{3+}(..)}. \quad (27)$$

To be able to easily see the working of this method, compare it with the KNS and MKNS models and to see its property of preserving the marginals of the original model, in Table 4 I present the joint and marginal counts of a sample bigram maximum likelihood model, smoothed with MDKNSPOMD, trained on the same small text as used in Tables 1, 2 and 3.

**Table 4.** Joint and marginal counts of a simple maximum likelihood model with MDKNSPOMD trained on the sample text.

c(x, y)	<s>	a	b	c	d	e	</s>	
<s>	0.00	2.04	2.15	4.15	0.46	1.45	0.76	11.00
a	0.00	3.94	2.16	3.70	1.90	7.47	1.84	21.00
b	0.00	6.30	1.89	1.60	0.46	0.82	2.94	14.00
c	0.00	2.43	4.49	2.23	0.62	3.28	1.95	15.00
d	0.00	1.35	0.62	0.62	1.21	0.52	1.67	6.00
e	0.00	4.94	2.70	2.70	1.35	5.47	1.84	19.00
</s>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	21.00	14.00	15.00	6.00	19.00	11.00	86.00

The smoothing method presented above for a bigram model can be generalized to higher-order models without a problem. To do this one has two

possibilities. First, one can simply use the above model on the highest level of the model and interpolate it only with the second-to-highest level, not smoothing that level further. This version has the advantage that all the original marginals are preserved. However, one has to note that in case there are many zeros in the original model (such as in n-gram language models with  $n \geq 3$ ), this solution will not work well as it will still leave too many zeros in the model.

The other solution, as also proposed for other smoothing techniques in the past, is to do the interpolation recursively, always interpolating the  $n^{th}$  level with the  $(n - 1)^{th}$  level, all the way back to the  $1^{st}$  or  $0^{th}$  level. In this case the theoretical and mathematical derivation previously applied on the bigram model for defining the right probability distributions in the lower level, can be used for all the levels. This would result in the following formula for a trigram model at the trigram level:

$$p_{MDKNSPOMD}(w_i|w_{i-2}w_{i-1}) = \frac{c(w_{i-2}w_{i-1}w_i) - D_3(c(w_{i-2}w_{i-1}w_i))}{\sum_{w_i} c(w_{i-2}w_{i-1}w_i)} + \gamma_{MDKNSPOMD}(w_{i-2}w_{i-1})p_{MDKNSPOMD}(w_i|w_{i-1}), \quad (28)$$

with  $D_3$  being a discount function similar to the original  $D$  discount function, with values 0,  $D_{(3;1)}$ ,  $D_{(3;2)}$  and  $D_{(3;3+)}$ , and the normalization factor being:

$$\gamma_{MDKNSPOMD}(w_{i-2}w_{i-1}) = \frac{D_{(3;1)}N_1(w_{i-2}w_{i-1}.)}{\sum_{w_i} c(w_{i-2}w_{i-1}w_i)} + \frac{D_{(3;2)}N_2(w_{i-2}w_{i-1}.) + D_{(3;3+)}N_{3+}(w_{i-2}w_{i-1}.)}{\sum_{w_i} c(w_{i-2}w_{i-1}w_i)}. \quad (29)$$

With the same logic one gets the bigram level:

$$p_{MDKNSPOMD}(w_i|w_{i-1}) = \frac{D_{(3;1)}N_1(.w_{i-1}w_i) + D_{(3;2)}N_2(.w_{i-1}w_i)}{D_{(3;1)}N_1(.w_{i-1}.) + D_{(3;2)}N_2(.w_{i-1}.) + D_{(3;3+)}N_{3+}(.w_{i-1}.)} + \frac{D_{(3;3+)}N_{3+}(.w_{i-1}w_i) - D_2}{D_{(3;1)}N_1(.w_{i-1}.) + D_{(3;2)}N_2(.w_{i-1}.) + D_{(3;3+)}N_{3+}(.w_{i-1}.)} + \gamma_{MDKNSPOMD}(w_{i-1})p_{MDKNSPOMD}(w_i), \quad (30)$$

with the normalization factor being:

$$\gamma_{MDKNSPOMD}(w_{i-1}) = \frac{D_2N_{N_{1+}}(.w_{i-1}.)}{D_{(3;1)}N_1(.w_{i-1}.) + D_{(3;2)}N_2(.w_{i-1}.) + D_{(3;3+)}N_{3+}(.w_{i-1}.)}. \quad (31)$$

Finally, the unigram level can be formulated as follows:

$$p_{MDKNSPOMD}(w_i) = \frac{N_{N_{1+}}(..w_i)}{N_{N_{1+}}(..)}, \quad (32)$$



with  $N_{N_{1+}}(..w_i)$ ,  $N_{N_{1+}}(.w_{i-1}.)$  and  $N_{N_{1+}}(...)$  defined as:

$$\begin{aligned} N_{N_{1+}}(..w_i) &= |\{w_{i-1} : N_{1+}(.w_{i-1}w_i) \geq 1\}|, \\ N_{N_{1+}}(.w_{i-1}.) &= |\{w_i : N_{1+}(.w_{i-1}w_i) \geq 1\}|, \\ N_{N_{1+}}(...) &= |\{w_{i-1}, w_i : N_{1+}(.w_{i-1}w_i) \geq 1\}|. \end{aligned} \quad (33)$$

With such a multilevel model, it is advantageous to set a different set of discount parameters at each level, based on the properties of that level (e.g. with the previously shown formulas applied to each level). However, please note that because of the non-integer values at the bigram level, it is not possible to use 3 different discounting parameters the same way as done at the trigram level, therefore only a single discount parameter ( $D_2$ ) is used at the bigram level.

There are a couple of further details to be considered. First, negative values have to be avoided at each level, which can be achieved by discounting in the form  $\max(0, c - D)$  instead of simple subtraction. Moreover, in case of the  $(n - 1)^{th}$  level, the basic values should represent the sum of the discounts truly subtracted at the  $n^{th}$  level for the given trigrams, considering the possibly reduced discount values due to the used  $\max$  function. To avoid over-complicated equations, these properties were not included in the above formulas.

My method can work very well for any model, even for ones with a huge number of zeros. However, I have to note that in case of the recursively interpolated version, it only preserves the marginal probabilities at the highest level (e.g. in case of a trigram model only the marginals in the form  $p(..w_i)$  are preserved, and the marginals in the form  $p(.w_{i-1}w_i)$  are not). Preserving the others is not possible in such a case, as there exists only one  $(n - 1)^{th}$  level probability distribution that preserves all the marginals, and that is exactly the one without further interpolation from it. Nevertheless, the MDKNSPOMD model still has better properties than the KNS and MKNS models in case of recursive interpolation, as neither of them keeps any of the original marginals in such a case, with the MKNS not keeping them in case of simple two-level smoothing either.

## 4 Evaluation Methodology and Results

To see how well my proposed MDKNSPOMD method performs compared to previous ones, I have chosen a standard n-gram language modeling evaluation task. I have used the British National Corpus (version 2; BNC, ~100M words)<sup>1</sup> and the text of the full English Wikipedia database dump of 01.12.2015 (EnWiki, ~2000M words)<sup>2</sup> to evaluate the models on, both of which I previously pre-processed. Among other pre-processing steps, all text was converted to be fully lowercase. Further, in case of an n-gram model, for each sentence I added n-1

<sup>1</sup> <http://www.natcorp.ox.ac.uk/>

<sup>2</sup> The plain text from the Wikipedia database dump was obtained with the help of the Wikipedia Extractor ([http://medialab.di.unipi.it/wiki/Wikipedia\\_Extractor](http://medialab.di.unipi.it/wiki/Wikipedia_Extractor)) by Giuseppe Attardi.

special characters at its beginning as sentence starting symbols ( $\langle s \rangle$ ,  $\langle s2 \rangle$ , etc.) and a special character at its end marking the end of the sentence ( $\langle /s \rangle$ ), to be able to fully evaluate all the meaningful words of the sentences.

In case of each corpus, I used the words occurring at least 10 times in it as vocabulary, resulting in a vocabulary size of  $\sim 100k$  in case of the BNC, and  $\sim 1.2M$  in case of the EnWiki corpus. Special and punctuation tokens were always considered as separate words. To achieve robust results, the average entropy and perplexity was computed using 10-fold cross-validation (inside the folds the evaluation was done sentence-by-sentence). All tests were conducted with a slightly modified version of the Kyoto Language modeling Toolkit (Kylm)<sup>3</sup>.

To be able to draw detailed conclusions, tests with both 2- and 3-gram models were conducted. In case of models above the bigram level there would be two possibilities, as noted before: to only smooth the highest level, namely only interpolating back to the second-to-highest level, or to interpolate each  $n^{th}$  level with the  $(n - 1)^{th}$  level recursively, all the way back to the  $1^{st}$  (unigram) level. However, as noted before, when the second-to-highest level is not smoothed further, then it leaves far too many zeros in a language model. This would result in zero probabilities for many sentences, making it impossible to use in practice for this type of task, and resulting in an entropy and perplexity of Infinity during evaluation. Because of this only the results for the variants that use smoothing at all levels are presented here.

All my results are shown in Table 5. These confirm previous findings that, with the use of multiple discount parameters, the MKNS slightly outperforms the original KNS in all the test cases, with both having a clear advantage over simple absolute discounting (Abs). Further, it comes as no surprise that in case of all smoothing methods, results on the 3-gram models are always remarkably better than on the 2-gram models.

Looking at the choice of the corpus, there is only a slight difference in the results in case of the 2-gram models. From this I assume that the much larger size of the EnWiki corpus is compensated by the fact that the BNC is a much more balanced corpus, containing only well-written and grammatically correct texts, and having a much narrower scope in terms of the topics covered. However, in case of the 3-grams the EnWiki corpus has a very clear dominance over the BNC, which is no surprise, as training a 3-gram model requires much more training data than a 2-gram model, so in this case clearly the relevance of the larger size of the training corpus becomes more important than the advantages of the BNC.

When comparing my results to that of the previous methods I can see that the MDKNSPOMD consistently outperforms the simple absolute discounting as well as the original KNS. It achieves approximately the same results as the MKNS, although there seem to be a consistent very small margin between them in favour of the MKNS in case of the 3-gram models.

Beside the evaluation in a standard language modeling task, I also plan to test my novel smoothing method in other applications too, including my methods

<sup>3</sup> <http://www.phontron.com/kylm/>

**Table 5.** Detailed results of the tested smoothing algorithms.

Method	Model	Corpus	Perplexity	Entropy
Abs	2-gram	BNC	252.15	7.96
		EnWiki	251.82	7.98
	3-gram	BNC	156.01	7.26
		EnWiki	113.94	6.83
KNS	2-gram	BNC	242.57	7.91
		EnWiki	246.43	7.94
	3-gram	BNC	139.46	7.10
		EnWiki	104.76	6.71
MKNS	2-gram	BNC	241.18	7.90
		EnWiki	245.96	7.94
	3-gram	BNC	136.82	7.08
		EnWiki	103.72	6.69
MDKNSPOMD	2-gram	BNC	241.17	7.90
		EnWiki	245.98	7.94
	3-gram	BNC	137.50	7.08
		EnWiki	104.18	6.70

for the automatic interpretation of noun compounds [4] and the automatic computation of semantic similarity of words [3].

## 5 Summary and Conclusions

Within this paper I have given a detailed overview of the motivation for smoothing methods and the most frequently used smoothing techniques. I have shown that, despite its excellent performance, the MKNS does not preserve the marginal distributions of the original data, which would be advantageous in many cases, and which, according to Goodman [6], would be a requirement for a smoothing method to be optimal under certain assumptions. To overcome this problem, I have shown a modified version of the MKNS, called the MDKNSPOMD, that leaves the original marginal distributions unchanged. I have also shown that this is the only possible way of achieving this.

Beside simple problems, this model can be generalized to higher-order models too. For problems with a fairly low number of zeros it is usually enough to smooth the highest level, in which case all the original marginals are preserved. If there are too many zeros, one has to recursively smooth the lower levels too, but this way only the marginals at the highest level are preserved (it is impossible to preserve the other marginals in such a case). Nevertheless, the MDKNSPOMD is still better than the KNS and MKNS methods here too, as neither of them would keep any of the original marginals at any of the levels in such a case.

To compare this novel smoothing method with previous techniques, thorough tests have been conducted on a standard language modeling task, using two different corpora and evaluating on both 2- and 3-gram models using 10-fold

cross-validation. The results show that the MDKNSPOMD performs better than both simple absolute discounting and the KNS in case of all settings, and achieves about the same results as the MKNS, with the MKNS seeming to have a minor superiority in case of the 3-gram models.

Based on this I can conclude that the novel MDKNSPOMD could be successfully used in any problem where smoothing is required, and should definitely be preferred over other methods in case preserving the marginal distributions is required or would be advantageous.

## References

1. Chen, S., Goodman, J.: An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* 13, 359–394 (1999)
2. Chen, S.F., Rosenfeld, R.: A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing* 8(1), 37–50 (2000)
3. Dobó, A., Csirik, J.: Computing semantic similarity using large static corpora. In: van Emde Boas et al. (ed.) *SOFSEM 2013: Theory and Practice of Computer Science*, LNCS, vol 7741. pp. 491–502. Springer, Berlin, Heidelberg (2013)
4. Dobó, A., Pulman, S.G.: Interpreting noun compounds using paraphrases. *Procesamiento del Lenguaje Natural* 46, 59–66 (2011), <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/842>
5. Good, I.J.: The population frequencies of species and the estimation of population parameters. *Biometrika* pp. 237–264 (1953)
6. Goodman, J.T.: A bit of progress in language modeling. *Computer Speech & Language* 15(4), 403–434 (2001)
7. Jelinek, F., Mercer, R.: Interpolated estimation of markov source parameters from sparse data. In: *Workshop on Pattern Recognition in Practice* (1980)
8. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: *International Conference on Acoustics, Speech, and Signal Processing*. pp. 181–184 (1995)
9. marquis de Laplace, P.S.: *Essai philosophique sur les probabilités*. Bachelier (1825)
10. Ney, H., Essen, U.: On smoothing techniques for bigram-based natural language modelling. In: *1991 International Conference on Acoustics, Speech, and Signal Processing*. pp. 825–828 (1991)
11. Roark, B., Allauzen, C., Riley, M.: Smoothed marginal distribution constraints for language modeling. In: *The 51nd Annual Meeting of the Association for Computational Linguistics*. pp. 43–52 (2013)
12. Siivola, V., Hirsimäki, T., Virpioja, S.: On growing and pruning Kneser-Ney smoothed  $n$ -gram models. *IEEE Transactions on Audio, Speech, and Language Processing* 15(5), 1617–1624 (2007)
13. Sundermeyer, M., Schlüter, R., Ney, H.: On the estimation of discount parameters for language model smoothing. In: *The 12th Annual Conference of the International Speech Communication Association*. pp. 1433–1436 (2011)
14. Teh, Y.W.: A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore (2006)
15. Witten, I.H., Bell, T.C.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory* 37(4), 1085–1094 (1991)

16. Zhang, H., Chiang, D.: Kneser-Ney smoothing on expected counts. In: The 52nd Annual Meeting of the Association for Computational Linguistics. pp. 765–774 (2014)



# Towards Topic Detection in Plain Documents Using Key Terminology Extraction Supported by Institute of Technology Tallaght

Juan Huetle-Figueroa<sup>1</sup>, Fernando Perez-Tellez<sup>1</sup>, David Pinto<sup>2</sup>

<sup>1</sup> Institute of Technology Tallaght,  
Ireland

<sup>2</sup> Benemérita Universidad Autónoma de Puebla,  
Mexico

{juan.huetle, fernandopt, juan.huetle, fernandopt}@gmail.com,  
dpinto@cs.buap.mx

**Abstract.** This paper presents an approach for helping in the topic detection tasks. The idea is to use collocation measures to extract key terminology from plain text. We use three measures for ranking N-grams (sequence of terms), Point mutual information, Likelihood-ratio and Chi-square. With this measures we built three different groups: bigrams, trigrams and quadrigrams. Each of the measures were implemented with the purpose of comparing and helping in the detection of good key terminology in plain text. In order to obtain the best N-grams, we have implemented two filters: the first one is to get common N-grams with the highest values in the three measures (intersection). For this, we use the most significant percentages of N-grams to create subsets and then select the key terminology with high value in the three measures. The second filter is to detect the occurrence of important collocations based on part-of-speech patterns. The corpora used in this research work was obtained from the website jobs, i.e. related to job descriptions. In the results we show the key terminology extracted by this approach to demonstrate its effectiveness.

**Keywords:** collocations, n-grams, POS, key term extraction.

## 1 Introduction

Collocations refer to words relationships, there are useful in the natural language processing area (NLP). They are expressions formed with two or more consecutive terms that correspond to a way of saying concrete ideas or concepts. They usually include noun phrases such as *deep learning* or phrasal verbs such as *to look for*. The use of collocations in the NLP area makes the text to sound natural and makes more sense to people. The importance of the experiments presented in this research work is to obtain a list of relevant topics discussed in plain text

through the detection of key terminology. In order to achieve this goal we have defined a range of measures to compare them with each other and detect the best key terminology in a given text.

The measures used are Pointwise mutual information (PMI), Likelihood-ratio and Chi-square. They were chosen for simplicity, low compute capacity required and they showed acceptable results in the experiments. Also in this research work, we used N-grams which are sequence of  $n$  terms, in particular we used bigrams (two terms), trigrams (three terms) and quadrigrams (four terms). We created two different experiments one with stop words and another without stop words. Stop words are words with very little or no lexical meaning such as *and*, *a*, *to*, and *in*). Therefore, we defined two different ways of analysing the N-grams. We considered that the phrase “state of the art” does not have the same meaning as “state art”. It can be noticed that these phrases have two different meanings. The first phrase refers to “the latest and most sophisticated or advanced stage of a technology, art, or science.”[12] and the second has no understandable meaning. Ranking N-grams with measures mentioned before we could understand and identify different key terminology. To do this, we extracted the best terms by selecting collocations with high value in the measures mentioned previously.

In this research work, we used intersection between the sets of N-grams ranked by collocation measures and filtered by the highest values. The intersection task is when a set of N-grams filtered by the highest value of a collocation measure appears in another set of N-grams filtered by a different collocation measure. In Table 6 the phrase “dublin city centre” appears with high value in PMI, Likelihood-ratio and Chi-square.

**Table 1.** Universal POS.

Universal POS	
ADJ: adjective	PART: particle
ADP: adposition	PRON: pronoun
ADV: adverb	PROPN: proper noun
AUX: auxiliary	PUNCT: punctuation
CCONJ: coordinating conjunction	SCONJ: subordinating conjunction
DET: determiner	SYM: symbol
INTJ: interjection	VERB: verb
NOUN: noun	X: other
NUM: numeral	

We carried out experiments where parts of speech (POS) are used. In the Table 1 and 2, we present the different POS used in this research work: *noun*, *pronoun*, *adjective*, *determiner*, *verb*, *adverb*, *preposition*, *conjunction*, and *interjection*. They were used to identify different lexical patterns in the N-grams. We briefly explain the experiments carried out in this research work.



- **Experiment 1:** N-grams are created with stop words such as (*a, at, of, the, etc.*) to created the specific correct key terminology.
  - Experiments with universal and normal POS.
  - We refer as 'universal POS' to the tokenization in the text with 17 possible tags shown in Table 1.
  - We refer to normal POS to the tokenization in the text with 45 possible tags shown in Table 2.
- **Experiment 2:** N-grams are created without stop words to be able to use them in queries or phrases, it is not the same meaning "deep learning" and "deep" or "learning" as you can see they have two different meanings, the first pair of terms together refer to the specific area of computing and the other two terms individually are only single words with different meanings.
  - Experiments with normal and universal POS

Table 1 and Table 2 show the different POS tags and their respective meanings. Table 1 shows 17 different tags. Having few tags made us limited ourselves in our experiments. In this research work, we needed use specific tags such as the adjective(JJ) and superlative(JJS) to apply a filter. Besides that we need to use the comma, explained in Section 3.1 Prepossessing.

Table 2 shows the tags used in this research work. There are 45 possible tags, giving us a closer approximation about the text and it provides with the necessary tools to apply the filter used in this research work.

**Table 2.** Normal POS.

Normal POS		
Punctuation Marks: “”, “.” “,”, “”, “(”, “)”, “,”, “”, “.”	NNP: noun, proper, singular	VBD: verb, past tense
CC: conjunction, coordinating	NNS: noun, common, plural	VBG: verb, present participle or gerund
CD: numeral, cardinal	PDT: pre-determiner	VCN: verb, past participle
DT: determiner	POS: genitive marker	VBP: verb, present tense, not 3rd person singular
EX: existential there	PRP: pronoun, personal	VBZ: verb, present tense, 3rd person singular
FW: foreign word	PRP\$: pronoun, possessive	WDT: WH-determiner
IN: preposition or conjunction	RB: adverb	WP: WH-pronoun
JJ: adjective or numeral	RBR: adverb, comparative	WRB: Wh-adverb
JJR: adjective, comparative	RBS: adverb, superlative	
JJS: adjective, superlative	RP: particle	
MD: modal auxiliary	SYM: symbol	
NN: noun, common	TO: "to" as preposition	
	VB: verb, base form	

The rest of this paper is organized as follows. The next section provides a review of related work to obtain key terminology, methods and application. In Section 3, we describe the measures used and intersection of the terms with their measures. Section 4 describes the corpus (dataset) used for the experiments. Section 5 contains a description about the preprocessing of the data, and the two experiments carried out in this research work. Finally, in the last Section, we conclude the paper and outlines future work directions.

## 2 Related Work

Our research goal is to obtain key terminology from plain documents, we have studied previous research works focused on keyword extraction. Researchers in [13] have reported the use of statistical methods and approaches such as simple statistics, linguistics, machine learning approaches. They extracted small set of units, composed of one or more terms, from a single document. They discussed about the extraction of small units sets, composed of one or more terms, from a single document. It is an important problem in Text Mining (TM), Information Retrieval (IR) and Natural Language Processing (NLP). Authors focused on the graph-based methods. They have compared methods with existing supervised and unsupervised methods. On the other hand, [8] used statistical methods with TFIDF (term frequency, inverse document frequency) they described the use of TFIDF in different parts of the plain document. For example, if a word appears sporadically in more than half of the document it is also considered as a keyword without taking into account the stop words. As well as multiple times in a single paragraph but not in the overall document TFIDF will not consider the word as keyword considering its low frequency.

In [7] authors used unsupervised approaches to automate the keyword extraction process from meeting transcript documents and they incorporated the use part-of-speech (POS) information in similar manner that we did. Then, they identified key-words using F-measure and a weighted score relative, giving them good results with TFIDF. The data that they used was *meeting recordings* converted into text.

The authors of [11] automatically generated a headline for a single document. They mixed sentence extraction and machine learning, their corpus were scientific articles. Another interesting approach is [1] they combine resources for lexical analysis such as electronic dictionary, tree tagger, WordNet, N-grams, POS pattern, resulting in a survey, they used different dataset the most relevant for us is the web pages, encyclopedia article, newspaper articles, journal articles and technical report. In [14] used *salience* rank in 500 news articles, the result was to improve the quality of extracted keyphrases and balance topic in corpus.

There is also some research in the field of real-time automatic speech recognition. In [4] authors applied keywords to formulate implicit queries to a just-in-time-retrieval system for use in meeting rooms.

## 3 Measures

We used three types of collocation measures to define the best filter in the N-grams. These measures were chosen for the easy implementation, good results and the low computing power needed with large volume of information, the following measures have been reported in [10].

- **PMI** Pointwise mutual information is a measure of association:

$$pmi(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)}, \quad (1)$$

$pmi(x; y)$  means the association between two terms (bigram), the first word is represented with  $x$  and the second word with  $y$ . It's a popular measure for the simply implementation and the good results.

- **Likelihood-ratio** We used "maximum Likelihood-estimation" to decide if there is a important contrast between the expected and the observed frequencies in bigrams, trigrams and quadrigrams. This measure expected two hypothesis  $L(H_1)$  and  $L(H_2)$  shown in the formula (2). The following formula describe the occurrence frequency of a bigram  $w^1w^2$ .

**Hypothesis 1.** The occurrence of  $w^2$  is independent of the previous occurrence of  $w^1$ :

$$P(w^2|w^1) = p = P(w^2|-w^1).$$

**Hypothesis 2.** It is a formalization of dependence which is good evidence for an interesting collocation:

$$P(w^2|w^1) = p_1 \neq p_2 = P(w^2|-w^1).$$

For  $p, p_1$  and  $p_2$  and write  $c_1, c_2$ , and  $c_{12}$  for the number of occurrences of  $w^1, w^2$  and  $w^1w^2$  in the corpus[10].

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)}, \quad (2)$$

$$= \log \frac{b(c_{12}, c_1, p)b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1)b(c_2 - c_{12}, N - c_1, p_2)}, \quad (3)$$

$$= \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p), \quad (4)$$

$$- \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2). \quad (5)$$

- **Chi-square** We used Chi-square with the same purpose that Likelihood ratio search important contrasts between the frequencies in bigrams, trigrams and quadrigrams, the formula (6) shown how work:

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad (6)$$

where  $i$  ranges over rows of the table,  $j$  ranges over,  $O_{ij}$  is the observed value for cell  $(i, j)$  and  $E_{ij}$  is the expected value.

### 3.1 Intersection

We implemented Likelihood-ratio positive, because we are only interested in positive results. A positive result means an estimate of the occurrence of an N-gram in the corpus and a negative result is the estimate that an N-gram does not occur in the corpus. We create a filter derived from the aforementioned measurements, we take the results of each one and we intersect them giving a subset. That is to say each one has its own range, so only took the best results of each one. We represent the set PMI as set  $A$ , Likelihood-radio as set  $B$  and Chi-square as set  $C$ . Thus we get the following intersections.

- In Table 3, we can observe  $A \cap B$  (see Fig. ??). This intersection between two sets of values PMI and Likelihood-ratio, where both have high values and we see the 10 first trigrams with the highest value. For do the intersection only 50% was taken that is to say one subset from  $A$  and another  $B$ . You can see the difference in the N-gram "competitive salary earn" has in PMI 331.944 higher than Likelihood-ratio with 21.049 Table 3.

**Table 3.** Sample trigrams filtered by the intersection  $A \cap B$ .

Trigram	Freq.	PMI	Likelihood-ratio
dublin city centre	53	2019.562	17.231
telecoms tech support	13	501.210	18.167
successful candidate joining	3	496.637	18.667
third level qualification	7	349.176	18.474
benefits competitive salary	3	341.853	18.110
<b>competitive salary earn</b>	2	331.944	21.049
fast paced environment	6	328.250	17.544
equal opportunities employer	6	316.0285	21.500
competitive salary gym	2	314.754	18.242
proven track record	6	306.108	21.363

- In Table 4, we can observe  $A \cap C$  (see Fig. ??). This intersection between two measures PMI and Chi-square, where both have high values and we see the 10 first trigrams with the highest value. In special the term "equal opportunities employer" start to obtain key terminology. If you compare Table 3 with Table 4 you will start to see deleted terms.

**Table 4.** Sample trigrams filtered by the intersection  $A \cap C$ .

Trigram	Freq.	PMI	Chi-square
dublin city centre	53	8154393.488	17.231
telecoms tech support	13	3826386.926	18.167
successful candidate joining	3	1258923.953	18.667
benefits competitive salary	3	856803.543	18.110
competitive salary earn	2	4349623.070	21.049
fast paced environment	6	1149592.709	17.544
<b>equal opportunities employer</b>	6	17803759.837	21.500
competitive salary gym	2	628854.620	18.242

- In Table 5, we can observe  $B \cap C$  (see Fig. ??). This intersection between two measures Likelihood-ratio and Chi-square, where both have highest values and we see the 10 first trigrams with the highest value. We can see that measure Chi-square delete terms because they do not exist in their subset.
- In Table 6, we can observe  $A \cap B \cap C$  (see Fig. ??). This intersection between three measures PMI, Likelihood-ratio and Chi-square. It is one of the main

**Table 5.** Sample trigrams filtered by the intersection  $B \cap C$ .

Trigram	Freq.	Likelihood-ratio	Chi-square
related locations dublin	61	2371.108	1662990.691
centre job description	24	2202.153	994671.932
south job description	20	2198.519	1503450.729
cork job description	14	2067.289	539817.166
limerick job description	9	2021.597	559561.417
dublin city centre	53	2019.562	8154393.488
waterford job description	6	1987.107	501852.430
laois job description	4	1967.965	502968.807
locations dublin city	31	1787.154	1764364.726

objectives of this research work, because we can observe how begin to filter the information. You can see the respective measure of each one. When comparing the Tables 3, 4 and 5, we see that the measure with the most delete terms was Chi-Square.

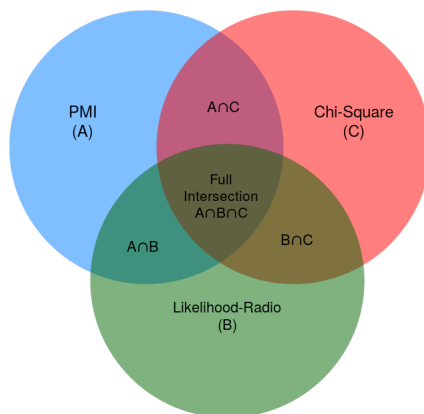
**Table 6.** Sample trigrams filtered by the intersection  $A \cap B \cap C$ .

Trigram	Freq.	PMI	Likelihood-ratio	Chi-square
<b>dublin city centre</b>	53	2019.562	8154393.488	17.231
telecoms tech support	13	501.210	3826386.926	18.167
successful candidate joining	3	496.637	1258923.953	18.667
third level qualification	7	349.176	2550273.661	18.474
benefits competitive salary	3	341.853	856803.543	18.110
competitive salary earn	2	331.944	4349623.070	21.049
fast paced environment	6	328.250	1149592.709	17.544
equal opportunities employer	6	316.028	17803759.837	21.500
competitive salary gym	2	314.754	628854.620	18.242
proven track record	6	306.108	16186542.685	21.363

## 4 Data

In this research work, we were working with jobs descriptions, all the data was taken from jobs.ie<sup>3</sup> a website in Ireland. The website has 46 different sectors and a number of jobs description on each sector. They are shown in the Table 7. Each job description file contains information as skills needed, payments and area of work. All the documents were in HTML and JSON format, we had to clean the documents from HTML tags, and download the updated information for each week. For this research work, we used in specific the IT (information technology) list count with 153 jobs descriptions, the average of clean files is 3

<sup>3</sup> <https://www.jobs.ie/>



**Fig. 1.** Set intersection.

Kilobytes per file. To collect these data we used a web crawler (HTTrack)<sup>4</sup> to automatically download all the jobs descriptions every week.

The reasons to chose these data are:

- The potential to use the key terminology to match job seeker and companies.
- The functionality of using different work sectors in the corpus.
- Use the N-grams in open questions for the companies.
- The volume of real information retrieved.
- The diversity of information content.
- To use the information obtained in the future in conjunction with the CV to make a semantic matches.

## 5 System Overview

We carrying out two different experiments: the first is using the stop words and part of speech and the second one was without stop words.

### 5.1 Preprocessing

The following list shows the preprocessing for this research work.

- We explained in section 4 that whole data was downloaded in HTML and JSON files.
- We clean all unnecessary lines such as HTML and JavaScript tags in the corpus.
- The information was stored in different files such as *job1*, *job2*, ... *jobn*.
- We created a string with all this information.

<sup>4</sup> <https://www.httrack.com>

**Table 7.** Categories of job descriptions.

Sector	num.	Sector	num.	Sector	num.
Academic	21	Pubs, Bars and Clubs	199	HR/Recruitment	102
Architecture/Design	20	Retail	293	Legal	52
Big Data/Business A.	17	Sales - Up to 35k	297	Manufact./Engineering	140
Chef Jobs	374	Security	34	Miscellaneous	54
Construction/Eng.	103	Telec./Tech Support	45	Multi-lingual	143
Education/Training	77	Travel/Tourism	92	Pharma./Sci./Agricul.	116
Financial Services	101	Warehouse/Logis./Ship.	153	Proper./Facilities Manag.	59
Franchise/Business	5	Accountancy/Finance	304	Restaurant/Catering	669
Hair and Beauty	100	Banking/Insurance	110	Sales - 35k+	270
Hotels	1021	Call-Centre/Cust. Serv.	340	Secre./Admin/PA	257
<b>IT</b>	153	Childcare	54	Senior Appointments	23
Manager/Supervisor	267	Drivers	71	Trades/Operative/Man.	144
Marketing	99	Renewable Energy	9	Charity Work	31
Motors	120	Fitness and Leisure	55	Work Exp./Internship	6
Online/Digital M.	47	Graduate	63		
Merchandising	43	Health/Med./Nursing	156		

- We removed all symbols such as @, ", ', \*, ?, , etc. because the job description is written by the companies and they usually use symbols.
- We convert all the letters in lowercase, because it is the same say "*computer science*" that "*Computer science*", only change the first letter and we had two different bigrams (in this case).
- We used NLTK<sup>5</sup> to tokenize the whole corpus with POS<sup>6</sup> functions, because NLTK works by context that is to say use the words before and after of each word, one example is "Support" could be a noun or verb.
- We discard possibles combinations with ". ", " ", " " and ";", for example we had a lot of incomplete ideas such as "*customers, and providing*" and "*innovation happens. And*". For this the program we developed uses a classification pattern when put a conditional.

For the second experiment we used a stop words list, to not discard combinations. In Table 8. we can see how was building the N-grams used in this research work.

## 5.2 Experiment 1

Experiment 1 presents the set intersection between the measures use to rank terms but without POS filter and order by Likelihood-ratio. We can see the different results in the Table 9 and 10.

<sup>5</sup> Natural Language Toolkit <https://www.nltk.org/>

<sup>6</sup> Part of speech

**Table 8.** How the N-grams were created.

N-gram	Freq.	N-gram	Freq.	N-gram	Freq.
hardware software	12	skills experience	16	software development	21
centre dublin	12	excellent communication	17	dublin city centre	21
dublin south job description	12	related job description	18	dublin city	24
city centre dublin	12	locations job description	19	city centre job description	24
part of team	13	related job	19	project management	24
team player	13	locations job	19	years experience	25
tech support	13	south job description	20	successful candidate	25
customer satisfaction	13	skills ability	20	related city	26
strong knowledge	13	south job	20	related city centre	26
work environment	14	locations city centre job	21	centre job	27

In Table 9, we can observe that the measures Chi-square and PMI are not congruent in a descending or ascending form. This is due to the fact that many terms were discarded by the intersection. The Likelihood-ratio results are ordered in a descending form but between each value there are a big difference, this is also due to the fact that N-grams were discarded.

To explain better why N-grams are discarded when the intersection of the three measurements is done. It is necessary to know that an intersection is a subset of other sets, in this case of three sets (measures). We call full intersection to this subset (see Fig. 1).

**Table 9.** Sample trigrams filtered by the intersection process.

Trigram	Freq.	Likelihood-ratio	Chi-square	PMI
related locations dublin	61	2371.108	1662990.691	14.732
centre job description	24	2202.153	994671.932	15.312
south job description	20	2198.519	1503450.729	16.178
cork job description	14	2067.289	539817.166	15.172
limerick job description	9	2021.597	559561.4171	15.856
dublin city centre	53	2019.562	8154393.488	17.231
waterford job description	6	1987.107	501852.430	16.271
laois job description	4	1967.965	502968.807	16.856
job description summary	3	1943.123	295844.586	16.441

### 5.3 Experiment 2

Experiment 2 is defined by the intersection of sets generated by the three collocation measures defined and a POS filter. We also used tokenization with POS tags. The POS filter consists in verify if the first word is tagged by a *JJ* or *NN* followed by any other tag or couple of tags and ending with a tag *NNS* or *NN*. For instance, in Table 10 we can see N-grams filtered by discarding mainly verbs.



In Table 11, we can observe the N-grams that did not followed the POS pattern defined. We can see a pattern at the beginning of the N-grams that start with the following tags: *IN*, *VB*, *VBG* or *RB*. Taking into account this pattern, the filter was created discarding all the N-grams that had that pattern. We called this discarding as POS filter.

It is important to note that we only defined the POS pattern at beginning and at the end of the N-grams that means that in the middle of the N-grams could be any other N-grams with any POS tag.

**Table 10.** Trigram with set intersection and filter with POS.

Trigram	Freq.	Likelihood-ratio	Chi-square	PMI
related/JJ locations/NNS dublin/NN	61	2371.108	1662990.691	14.732
centre/NN job/NN description/NN	24	2202.153	994671.932	15.312
south/NN job/NN description/NN	20	2198.519	1503450.729	16.178
cork/NN job/NN description/NN	14	2067.289	539817.166	15.172
limerick/NN job/NN description/NN	9	2021.597	559561.417	15.856
dublin/NN city/NN centre/NN	53	2019.562	8154393.488	17.231
waterford/NN job/NN description/NN	6	1987.107	501852.430	16.271
laois/NN job/NN description/NN	4	1967.965	502968.807	16.856
job/NN description/NN summary/NN	3	1943.123	295844.586	16.441
wicklow/NN job/NN description/NN	3	1939.786	242438.628	16.119

**Table 11.** Trigram with set intersection and tokenized but without filter POS.

Trigram	Freq.	Likelihood-ratio	Chi-square	PMI
ensure/VB customer/NN satisfaction/NN	2	223.515	41012.619	14.247
across/IN multiple/NN projects/NNS	2	208.225	69104.106	15.030
establish/VB best/JJS practice/NN	2	177.990	3266621.424	20.638
across/IN multiple/NN time/NN	2	176.725	92012.22154	15.458
rewarding/VBG work/NN environment/NN	3	170.576	191825.611	15.962
privately/RB owned/VBN media/NNS	3	156.291	67784782.307	24.429

## 6 Conclusion

We start out by choosing three measures: PMI, Chi-square and Likelihood-ratio to rank N-grams (bigrams, trigrams and quadrigrams) and obtain key terminology from different plain documents. We have shown that intersecting the highly ranked N-grams (with collocation measures) can help in filtering out irrelevant terms and identify useful key terminology. In this research work, we also have used specific POS tags to rule out the unnecessary N-grams. The POS pattern used to detect important key terminology consist of having the first word tagged

as *JJ* or *NN* followed by any word or couple of word with any POS tag(s) and ending with a word tagged as *NNS* or *NN*. This pattern contributed to obtain good results. This idea can be applied in other corpus to obtain key terminology by defining a POS pattern to filter relevant N-grams.

In these experiments, we show that using the POS patterns can help in better detection of key terminology. We also used the intersection of highly ranked N-grams by collocation measures and we got better key terminology when we applied both. Future work includes corpus evaluation with precision and recall to obtain the relevant subsets. We are also planning to use a thesaurus to enrich the key terminology obtained in this work then to use machine learning algorithms fed by the enriched key terminology.

## References

1. Bharti, S.K., Babu, K.S.: Automatic keyword extraction for text summarization: A survey (2017)
2. Bird, S.: NLTK: The natural language toolkit. In: Proceedings of the COLING/ACL on interactive presentation sessions, pp. 69–72, Association for Computational Linguistics (2006)
3. Brezina, V., McEnery, T., Wattam, S.: Collocations in context. *International Journal of Corpus Linguistics*, 20(2), pp. 139–73 (2015)
4. Habibi, M., Popescu-Belis, A.: Keyword extraction and clustering for document recommendation in conversations. *IEEE Trans Audio Speech Lang Process* 23(4), pp. 746–759 (2015)
5. Jurafsky, D., Martin, J.H.: Speech and language processing. Pearson (2014)
6. Kantor, P.: Foundations of statistical natural language processing. *Information Retrieval*, 4(1), pp. 80–81 (2001)
7. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proceedings of human language technologies: The Annual conference of the North American chapter of the association for computational linguistics, pp. 620–628, Association for Computational Linguistics (2009)
8. Luthra, S., Arora, D., Mittal, K., Chhabra, A.: A Statistical Approach of Keyword Extraction for Efficient Retrieval. *International Journal of Computer Applications*, 168(7) (2017)
9. Maldonado-Guerra, A., Emms, M.: Measuring the compositionality of collocations via word co-occurrence vectors: Shared task system description. In: Proceedings of the Workshop on Distributional Semantics and Compositionality (ACL), pp. 48–53, Portland (2011)
10. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT press (1999)
11. Mondal, A.K., Maji, D.K.: Improved algorithms for keyword extraction and headline generation from unstructured text. First Journal publication from SIMPLE groups, *CLEAR Journal* (2013)
12. Oxford dictionaries: <http://www.dictionary.com/browse/state-of-the-art?s=t> (2018)
13. Beliga, S.: Keyword extraction: A review of methods and approaches (2014)

14. Teneva, N., Cheng W., Saliency, R.: Efficient Keyphrase Extraction with Topic Modeling. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (2), pp. 530–535 (2017)



# El método de anagramas: un rápido y novedoso algoritmo para generar jugadas de Scrabble

Alejandro González Romero<sup>1</sup>, René Alquézar Mancho<sup>1</sup>,  
Arturo Ramírez Flores<sup>2</sup>, Francisco González Acuña<sup>2,3</sup>, Ian García Olmedo<sup>4</sup>

<sup>1</sup> Universidad Politécnica de Cataluña,  
Departamento de Ciencias de la Computación, Barcelona,  
España

<sup>2</sup> Centro de Investigación en Matemáticas, Guanajuato,  
México

<sup>3</sup> Universidad Nacional Autónoma de México,  
Instituto de Matemáticas,  
México

<sup>4</sup> Universidad Nacional Autónoma de México,  
Dirección General de Servicios de Cómputo Académico,  
México

yarnalito@gmail.com, alquezar@cs.upc.edu, ramirez@cimat.mx,  
ficomx@yahoo.com.mx, ian.garcia@gmail.com

**Resumen.** Todos los motores que juegan Scrabble necesitan un algoritmo para generar todas las jugadas legales; después, de alguna manera, se necesita seleccionar una jugada. Para escoger la mejor jugada es conveniente simular un cierto número de jugadas. Debido a las limitaciones de tiempo en un juego de torneo de Scrabble (30 minutos para todo un juego por jugador), no es práctico simular todas las jugadas legales; en nuestros experimentos el promedio de jugadas legales por turno fue de 971. Llamemos *candidatos* a las jugadas que serán simuladas en el futuro; si tenemos más candidatos tendremos una mayor probabilidad de escoger la mejor jugada. Este artículo presenta un rápido y novedoso algoritmo para generar todas las jugadas legales, el cual es usado por nuestro motor de Scrabble llamado *Heuri* [2,6]. El método está basado fuertemente en anagramas; este método tiene la belleza de imitar la manera en que los humanos buscan una jugada válida. Definiremos un *anagrama* de una cadena de caracteres como una palabra contenida en el lexicón que es obtenida por una permutación de los caracteres que forman la cadena. Ya que los anagramas son el alma del algoritmo lo llamaremos *El Método de Anagramas*. Además de presentar el método de anagramas, este artículo da una breve descripción de cómo *Heuri* juega y sobre los algoritmos usuales utilizados para generar jugadas legales. Estos algoritmos son utilizados por otros motores de Scrabble como *Quackle* [4]. Finalmente se dan tiempos de desempeño y comparaciones entre los algoritmos que

generan jugadas de Quackle y Heuri.

**Palabras clave:** anagramas, el método de anagramas, lexicón, lexicón de computadora, generación de jugadas válidas, Scrabble, motores de Scrabble.

## The Anagram Method: A Fast and Novel Scrabble Move Generator Algorithm

**Abstract.** All Scrabble engines need an algorithm to generate all legal moves, then somehow they have to select one move to be played. In order to select the best move it is convenient to simulate a certain number of candidate moves. Due to time limitations in a game of Scrabble it is impractical to simulate all legal moves; in our experiments the average number of legal moves per turn was 971 moves. Let us call *candidates* the moves that will be simulated; the more candidates we have the more likely to choose the best move. This paper presents a fast and novel algorithm to generate all legal moves; the algorithm is used by our Scrabble engine named *Heuri* [2,6]. The method is strongly based on anagrams; the beauty of this method is that it mimics the way humans search for a valid move. An *anagram* of a string here is a word contained in the lexicon that is obtained by a permutation of the tiles of the string. Since the anagrams are the soul of the algorithm we will call this method the *Anagram Method*. Besides presenting the Anagram Method, the paper gives a brief description of how Heuri plays and gives some information about the usual algorithms to generate legal moves employed by other Scrabble engines like *Quackle* [4]. Finally time performances and comparisons between the algorithms to generate moves used by Quackle and Heuri are given.

**Keywords:** anagrams, the anagram method, lexicon, computer lexicon, generation of valid moves, Scrabble, Scrabble engines.

### 1. Introducción

Estudiamos el Scrabble, un juego de palabras en el cual los jugadores hacen puntos colocando fichas con letras sobre un tablero dividido en  $15 \times 15$  cuadros. Las fichas deben formar palabras como en un crucigrama; para más información acerca del juego véase [8, 9].

Sobre la generación de jugadas válidas, Appel y Jacobson [1] introdujeron un algoritmo, el cual fue el más rápido y eficiente en su tiempo. Está basado en la estructura de datos DAWG (Directed Acyclic Word Graph, es decir, Gráfica Áciclica Dirigida de Palabras) construida a partir de las entradas de un lexicón. Después, Steve Gordon [3] introdujo una variante de esta estructura de datos GADDAG, la cual requiere 5 veces más espacio que el DAWG, pero duplica la velocidad de generación de jugadas.

En los ochentas, cuando las computadoras no tenían mucha memoria, el uso de DAWG para guardar el lexicon fue ingenioso y ahorró memoria. Sin embargo la memoria de las computadoras de hoy en día nos permite guardar los lexicones en listas que usan mucho más memoria que los DAWG y los GADDAG, pero este tipo de almacenaje, hecho de forma apropiada, aumentará la velocidad de generación de jugadas.

Hemos desarrollado Heuri [2, 6], un motor para jugar Scrabble que usa el método de anagramas para generar todas las jugadas legales. Heuri derrotó a un campeón mundial 6-0. Expliquemos cómo Heuri juega Scrabble:

Dados un tablero y un atril Heuri produce, para cada subatril del atril, todas las jugadas legales usando el método de anagramas (ver 3.2); además de generar las jugadas legales, Heuri también evalúa la puntuación de cada jugada (para una evaluación rápida cada cuadro guarda información como el puntaje hacia el norte, sur, oeste y este y qué tipo de premio tiene el cuadro).

Una parte importante de un programa que juegue Scrabble es la decisión de qué fichas dejar en el atril. En una jugada se juegan o cambian  $t$  fichas y las restantes  $n - t$  fichas conforman el *residuo* ( *leave* or *residue* ) ( $n = 7$  salvo quizá en el final).

Se da a todo residuo un valor como se describe en [6] o [2]. Es importante recordar que Heuri utiliza la posición actual del tablero para evaluar todo residuo. Heuri también calcula el puntaje de la jugada sobre el tablero. El valor del residuo y el puntaje de la jugada sirven para evaluar la jugada (véase [2, 6]).

Una vez que todas las jugadas son evaluadas y ordenadas, Heuri podría escoger, digamos, los primeros 40 candidatos. Entonces podríamos recalcular el valor de estos candidatos simulando cada uno de ellos; para tal efecto se encontrarían todas las posibles respuestas del oponente, usando 100 atriles aleatorios del adversario. Un algoritmo rápido para generar jugadas, como el método de anagramas, resulta esencial para esta fase. Esto mejoraría la defensa de Heuri tratando de evitar jugadas de alta puntuación del oponente.

Quackle también evalúa los residuos de cada posible subatril, pero a diferencia de Heuri, Quackle no utiliza el tablero real para estas evaluaciones; en vez de ello, Quackle precalcula cada posible residuo jugando cientos de miles de partidas. Por ello, cuando se juega en un nuevo lenguaje, para emplear toda su fuerza, Quackle necesita mucho tiempo para precalcular los nuevos residuos y Heuri solo necesita alrededor de 3 minutos para tener todo el lexicon de computadora listo y así poder jugar con toda su fuerza. Además, debido a que Heuri utiliza el tablero real para evaluar un residuo y Quackle no, los valores de los residuos de Heuri son más robustos que los de Quackle.

## 2. Algunas reglas de Scrabble y definiciones

### 2.1. El tablero de Scrabble

Un tablero estándar de Scrabble consiste en un conjunto de celdas que forman una cuadrícula. El tablero es de  $15 \times 15$  celdas o cuadros. Las fichas usadas en

el juego encajan en cada celda del tablero. El tablero de Scrabble denota sus columnas con etiquetas de la A hasta la O y sus renglones con etiquetas del 1 al 15. Este etiquetado nos permite referirnos a cuadros y palabras en el tablero. Hay cuadros con premio, usualmente denotados con diferentes colores de acuerdo a su tipo. Ellos se muestran en la Fig. 1.

Más formalmente, el tablero vacío es  $\beta = \{1, 2, \dots, 15\} \times \{1, 2, \dots, 15\}$ , que también denotaremos por  $\{1, 2, \dots, 15\} \times \{A, B, \dots, O\}$  (véase Fig. 1). Los elementos de  $\beta$  se llaman celdas o cuadros. A veces denotamos un cuadro por un número precedido de una letra o una letra precedida de un número. Por ejemplo,  $(8, 8)=8H=H8$ .

Un *tablero jugado* es un subconjunto  $T$  de  $\beta$  junto con una función que asigna a cada elemento de  $T$  una letra del alfabeto usado  $\{A, B, \dots, Z\}$ ; al subconjunto  $T$  solo también se le llamará tablero jugado. Cuando se juega Scrabble un *lexicón* es utilizado, el cual es la colección de palabras válidas.

Para cada carácter  $\lambda$  la bolsa inicial (que consiste de 100 fichas) contiene exactamente  $b(\lambda)$   $\lambda$ 's (ver [7] para más información acerca de las distribuciones de letras en Scrabble).

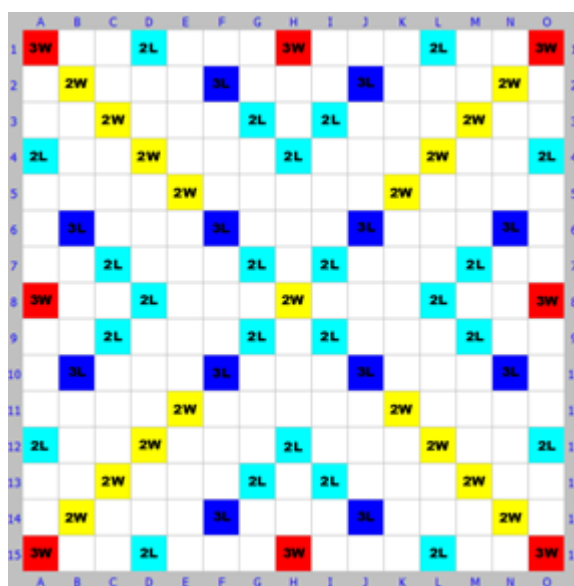


Fig. 1. Tablero de Scrabble.

En cada momento el jugador en turno tiene una colección no vacía de fichas (usualmente 7) que representan caracteres de  $\{A, B, \dots, Z, \#\}$  ( $\#$  denota un comodín y debe ser sustituido por una letra al jugarse en el tablero); estos caracteres forman un multiconjunto (un conjunto con posibles repeticiones; véase Knuth p. 694 [5]) al cual llamaremos un *atril* (*rack*).



## 2.2. Notación de palabras y opciones en un turno

Definamos una *cadena situada* como una palabra horizontal o vertical de longitud mayor que uno junto con las coordenadas de su carácter inicial. Una cadena situada no es necesariamente una palabra válida. Por ejemplo 8D FORMAL es una palabra horizontal que empieza en el cuadro 8D; H4 FORMAL es una palabra vertical que empieza en el cuadro H4. Ambas son cadenas situadas; también lo es H4 FRMLOA aunque FRMLOA no es una palabra legal.

Al principio cada jugador saca 7 fichas de la bolsa. Al terminar de hacer una jugada en el tablero (o cambiar), el jugador saca fichas de la bolsa de tal manera que el atril tenga 7 fichas (o menos si la bolsa no tiene suficientes fichas).

Hay tres opciones en cualquier turno. El jugador puede colocar una palabra, cambiar fichas por nuevas fichas o pasar.

Un *bingo* (o *scrabble*) es una jugada en la cual un jugador pone todas sus 7 fichas en el tablero haciendo una jugada válida; un *bingo* es premiado con un bonus adicional de 50 puntos. Ver [8, 9] para más información sobre el juego de Scrabble.

## 2.3. Algunos conceptos útiles

Definamos algunos conceptos que resultarán útiles para la explicación y entendimiento de cómo funciona el generador de jugadas de Heuri.

Sea  $S$  un subconjunto de  $\beta$ . Un cuadro  $\sigma$  de  $\beta$  es *adyacente* a  $S$  si no pertenece a  $S$  y un lado de  $\sigma$  es un lado de un cuadro  $\sigma'$  de  $S$ , es decir,  $|i - i'| + |j - j'| = 1$ , donde  $\sigma = (i, j)$  y  $\sigma' = (i', j')$ .

Sea  $T$  un tablero jugado. Si  $T \neq \emptyset$ , el *halo* (de  $T$ ) es el conjunto de cuadros adyacentes a  $T$ ; si no el *halo* es  $\{8H\}$ . Ver Fig. 2 para observar geométricamente el halo de cierto tablero jugado. El halo consiste en el conjunto de cuadros vacíos en la región encerrada por el polígono.

Sean  $L$  un renglón o columna de  $\beta$ ,  $T$  un tablero jugado y  $r$  la cardinalidad del atril en juego. Un *intervalo* en  $L$  es un subconjunto  $I$  de  $L$ , que interseca el halo, formado al menos por dos cuadros consecutivos de  $L$ , tal que ningún cuadro de  $T \cap L$  es adyacente a  $I$  y el número de cuadros vacíos de  $I$ ,  $|I - T|$ , es menor igual que  $r$ .

Intuitivamente un *intervalo* es una región en el tablero donde una palabra podría encajar o entrar. Una jugada válida hecha sobre el tablero debe ocupar un intervalo, pero existen intervalos donde no es posible hacer una jugada.

Damos algunos ejemplos, denotando un conjunto de cuadros consecutivos  $\{\sigma_1, \dots, \sigma_k\}$  por  $[\sigma_1, \sigma_k]$ . Supongamos que el tablero jugado  $T$  es el mostrado en la Fig. 2 y la cardinalidad  $r$  del atril en juego es 7. Tenemos que [10A,10K], [10D,10L], [10I,10O], [D4,D5], [N8,N10] son intervalos pero los siguientes no lo son: [11L,11N], [5E,5K] (no intersecan el halo); [D5,D9], [B10,B15] (son adyacentes a D10 y B9 respectivamente); [D6,D15] (contiene más de  $r$  cuadros vacíos).

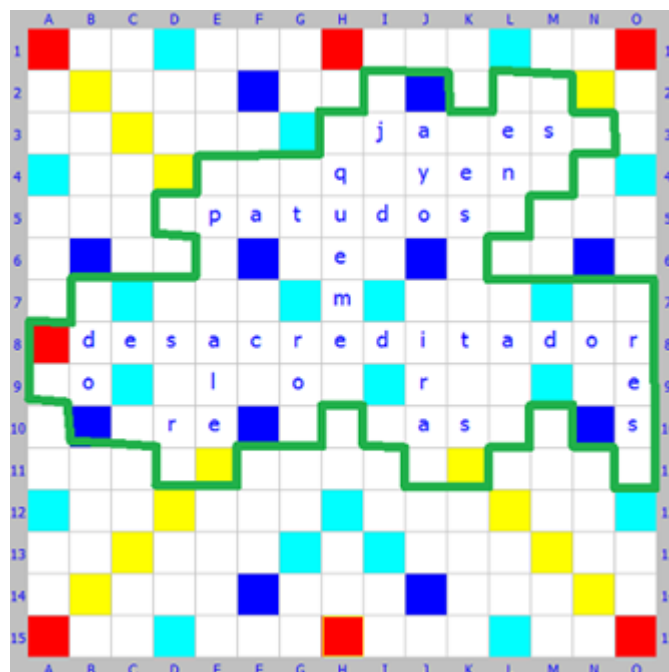


Fig. 2. Halo del Tablero Jugado.

## 2.4. Cálculo de palabras válidas

De las reglas de Scrabble, si el tablero no está vacío, para poder colocar una palabra, esta tiene que estar conectada al menos con una letra del tablero. Pero esto no es suficiente; para que una jugada sea válida, todas las palabras que se formen tienen que ser palabras válidas contenidas en el lexicon (desde el punto de vista de Heuri, contenidas en el lexicon de la computadora). Recuérdese que la dirección de juego en Scrabble es de norte a sur y de oeste a este.

Sean  $m$  y  $n$  cadenas; entonces  $mn$  indicará la concatenación de  $m$  y  $n$ . Para cada cuadro vacío del tablero se colecciona información acerca de las fichas jugadas arriba, abajo, a la izquierda y derecha del cuadro.

Sean  $T$  un tablero jugado y  $\sigma \in \beta$  un cuadro vacío ( $\sigma = (i, j) \notin T$ ). Si  $(i-1, j) \in T$ , defínase  $n_\sigma$  (el norte de  $\sigma$ ) como la cadena situada máxima que ocupa cuadros consecutivos  $(k, j)$  de  $T$  con  $k < i$ , e  $(i-1, j) \in n_\sigma$ ; de lo contrario  $n_\sigma$  es la cadena vacía. En una forma similar se definen  $s_\sigma$ ,  $w_\sigma$ , y  $e_\sigma$ , el sur, oeste y este de  $\sigma$ .

Ahora, si  $n_\sigma$  o  $s_\sigma$  es no vacía, defínase  $H_\sigma$  como el conjunto de letras  $\lambda$  tales que la concatenación  $n_\sigma \lambda s_\sigma$  es una palabra del lexicon; si no  $H_\sigma$  es todo el alfabeto; en este artículo los alfabetos utilizados consisten en el conjunto de letras  $\{A, B, \dots, Z\}$ . De forma similar, se define  $V_\sigma$  si  $w_\sigma$  o  $e_\sigma$  es no vacío; si  $w_\sigma = e_\sigma = \emptyset$ ,  $V_\sigma$  es todo el alfabeto.

Nótese que si  $\sigma$  es un cuadro vacío que no está en el halo de  $T$  entonces  $H_\sigma$  y  $V_\sigma$  es todo el alfabeto.

Si  $\sigma \in T$  entonces se define  $H_\sigma$  y  $V_\sigma$  como el conjunto de una sola letra, a saber, aquella representada por la ficha en  $\sigma$ .

Una palabra en el lexicón que ocupa las celdas consecutivas  $\sigma_1, \dots, \sigma_r$  de un renglón (resp. una columna) es una jugada válida si y solo si la letra representada por la ficha en  $\sigma_i$  ( $i \in [1, r]$ ) pertenece a  $H_{\sigma_i}$  (resp.  $V_{\sigma_i}$ ) y alguna  $\sigma_i$  pertenece al halo.

Si  $T$  es el tablero vacío el halo es la celda o cuadro 8H. Por tanto, cualquier palabra válida horizontal o vertical que contenga a la celda o cuadro central es una jugada válida.

Llamaremos a  $H_\sigma$  (resp.  $V_\sigma$ ) *el conjunto de letras admisibles horizontalmente* (resp. *verticalmente*) en  $\sigma$ , o brevemente, *el conjunto horizontal* (resp. *vertical*) de  $\sigma$ .

Para producir jugadas horizontales (resp. verticales) se usa  $H_\sigma$  (resp.  $V_\sigma$ ). Menos intuitivas son las jugadas de una dirección de una ficha, ya que  $H_\sigma$  (resp.  $V_\sigma$ ) corresponde a la construcción vertical (resp. horizontal) de una palabra. Las jugadas de dos direcciones de una ficha forman dos palabras (una horizontal y otra vertical).

Los siguientes ejemplos muestran como se usan los intervalos y los conjuntos horizontales y verticales de  $\sigma$  ( $H_\sigma$  y  $V_\sigma$ ), para colocar palabras válidas en el tablero.

Supongamos que un jugador tiene el atril { D E E I M N T } y el tablero mostrado en la Fig. 3 de una partida jugada en español. Las letras en cuadros ocupados se escribirán dentro de paréntesis. Si  $\sigma$  es un cuadro desocupado del renglón 10 entonces  $H_\sigma$  es el conjunto de todas las letras {A,B,...,Z} con las siguientes excepciones:

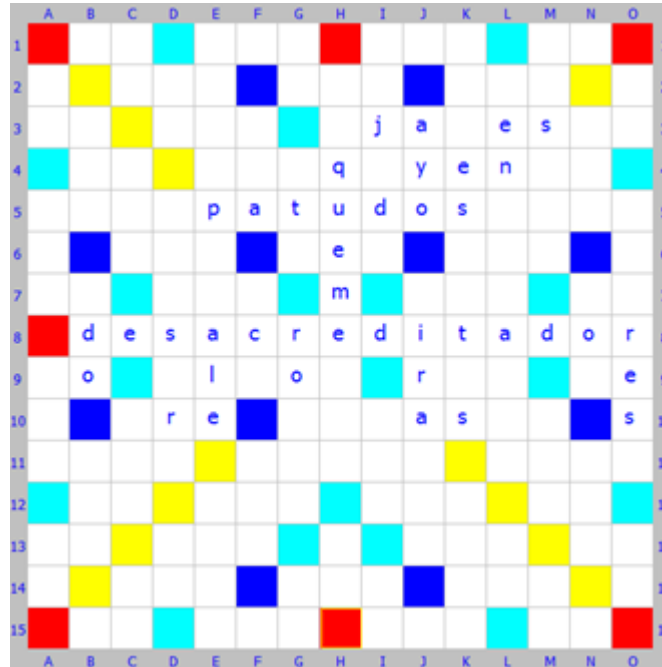
Si  $\sigma = 10B$  (el segundo cuadro del renglón 10) entonces  $H_\sigma = \{M, N, S, Y\}$  ya que B8 (DO) $\lambda$  es una palabra válida si y solo si  $\lambda$  pertenece a {M,N,S,Y}, y si  $\sigma = 10G$  (el séptimo cuadro) entonces  $H_\sigma = \{A, B, E, I, L, N, O, S\}$  ya que G8 (RO)A, (RO)B, (RO)E, (RO)I, (RO)L, (RO)N, (RO)O, (RO)S son palabras válidas y (RO) $\lambda$  no lo es si  $\lambda \notin \{A, B, E, I, L, N, O, S\}$ .

Algunas palabras que se pueden jugar en el renglón 10 son: 10C I(RE) (una jugada de una letra que contiene RE), 10D (RE)MEND(AS)TEI(S) (un bingo que contiene RE, AS y S), 10A ENT(RE)MEDI(AS) (un bingo que contiene RE y AS), 10I M(AS)EEI(S) (que contiene AS y S), 10B MI(RE)N (contiene RE), 10H ME(AS)EN (contiene AS), 10M DE(S) (que contiene S).

También 10A EN (en el intervalo que precede a RE) y 10G ET (en el intervalo entre RE y AS) son jugadas válidas.

En la columna E, si  $\sigma$  es un cuadro sin ocupar entonces  $V_\sigma$  es el conjunto de todas las letras {A,B,...,Z}, sin excepciones, ya que, cuando se juegan fichas solo en la columna E, no se forman palabras horizontales.

Si  $\sigma$  es un cuadro sin ocupar de la columna K, entonces  $V_\sigma$  es el conjunto de todas las letras {A,B,...,Z} con las siguientes excepciones:



**Fig. 3.** Una posición de Scrabble.

Si  $\sigma = K3$  entonces  $V_\sigma = \{ D, L, M, S, T \}$  ya que  $3I (JA)\lambda(ES)$  es una palabra válida si y solo si  $\lambda \in \{ D, L, M, S, T \}$ ; y si  $\sigma = K9$  entonces  $V_\sigma = \{ E, O \}$  ya que  $(9J (R)E, 9J (R)O)$  son las únicas palabras válidas de la forma  $(R)\lambda$ .

Algunas palabras verticales que se pueden jugar son:  $F4 M(A)TI(C)E$  sobre el intervalo  $[F4, F9]$  ( $9E (L)E(O)$  es una palabra válida),  $K3 D(ES)TE(T)E(S)$  en  $[K3, K10]$  ( $3J (JA)D(ES)$  y  $9J (R)E$  son válidas ).

Un par de palabras horizontales que usan intervalos que tienen intersección vacía con el tablero jugado son:  $2J MIDEN$  en  $[2J, 2N]$  (una jugada válida ya que  $2J MIDEN, J2 M(AYO), L2 D(EN), M2 E(S)$  son válidas) y  $7J MEDI$  en  $[7J, 7M]$  (válida ya que  $7J MEDI, J7 M(IRA), K7 E(T), L7 D(A), M7 I(D)$  son todas válidas). Véase la Fig. 3.

### 3. El generador de jugadas de Heuri: Estrategia anagramática

#### 3.1. El lexicón de la computadora

Una vez que se tiene un lexicón en un idioma dado, las palabras del lexicón se arreglan en distintos archivos: Directos, Inversos, Indices y Anagramas. Algunos de estos archivos serán usados por el generador de jugadas de Heuri para producir

jugadas que pueden hacerse, y otros se usarán para calcular los valores de las jugadas.

**Archivos directos e inversos.** Los archivos  $Directos_n$  e  $Inversos_n$  consisten en listas que tienen todas las palabras válidas de longitud  $n$  ( $2 \leq n \leq 15$ ), en orden directo o inverso. Nótese que, en Scrabble, no hay palabras de longitud uno. Se usan  $Directos_n$  e  $Inversos_n$  para calcular más eficientemente  $H_\sigma$  y  $V_\sigma$  (al buscar palabras,  $Directos_n$  es un poco más rápido que  $Inversos_n$  si el prefijo de la palabra buscada es más largo que el inverso del sufijo) y también se usan para evaluar las palabras perpendiculares.

Para calcular  $H_\sigma$  (resp.  $V_\sigma$ ) cuando la longitud de  $s_\sigma$  es mayor que la de  $n_\sigma$  (resp. la longitud de  $e_\sigma$  es mayor que la  $w_\sigma$ ), se usa  $Inversos_n$  donde  $n - 1$  es la suma de las longitudes de  $n_\sigma$  y  $s_\sigma$  (resp.  $e_\sigma$  y  $w_\sigma$ ). Si no es así, se usa  $Directos_n$ .

**Archivos de índices y anagramas.** Los archivos de índices contienen colecciones de cadenas que, después de substituir #'s por letras adecuadas, forman, aplicando una permutación, al menos una palabra válida. Estas cadenas también contienen un apuntador al final, que indica el principio de todos los anagramas que corresponden a cada cadena.

Los archivos de anagramas contienen todos los anagramas de una cadena dada. Un anagrama de una cadena  $w$  es una palabra válida obtenida reemplazando #'s en  $w$  (si existen) por letras y volviendo a arreglar la cadena resultante. Todo miembro (cadena) de  $Indices_n$  está convenientemente ligado a un conjunto de anagramas de  $Anagramas_n$  ( $n$  es la longitud de la cadena o palabra).

Los archivos de índices y anagramas son el alma del generador de jugadas. Véase la siguiente sección para mayor información.

Al principio de todo juego los archivos  $Directos_n$ ,  $Inversos_n$ ,  $Indices_n$ , y  $Anagramas_n$  ( $2 \leq n \leq 15$ ) se leen una vez de disco y se guardan en RAM, usando arreglos y tablas hash donde los  $Indices_n$  son las claves y los  $Anagramas_n$  son los valores.

**Construcción de archivos de índices y anagramas.** Los archivos de índices se obtienen usando todas las palabras válidas que están contenidas en los archivos directos, reemplazando 0,1 o 2 letras por comodines ( # ) en cada miembro de los archivos directos y finalmente ordenando lexicográficamente cada miembro y la colección de todos los miembros.

Describamos, de manera más precisa, la construcción de los archivos.

1. Si  $\lambda$  es una letra que aparece exactamente  $n(\lambda)$  veces en una cadena  $u$  y  $b(\lambda)$  es el número de  $\lambda$ 's en la bolsa inicial, definimos  $e(u)$ , el *exceso* de  $u$ , por  $e(u) = \sum \max \{0, n(\lambda) - b(\lambda)\}$  donde  $\lambda$  recorre el conjunto de letras que aparecen en  $u$ . Por ejemplo,  $e(\text{safaris})=0$ ,  $e(\text{maximum})=1$  y  $e(\text{ñiquiñaque})=2$ .

Para una palabra  $w \in Direct_n$  sea  $\{w'_1, \dots, w'_{m(w)}\}$  el conjunto de cadenas distintas obtenidas como sigue: sustitúyanse 0, 1 o 2 letras en  $w$  por #'s, y ordénese lexicográficamente cada cadena resultante (el # viene después de las

letras); se requiere que el número de  $\#$ 's en cada  $w'_i$  sea menor o igual que el número no negativo  $2 - e(w'_i)$ . Se puede calcular  $m(w)$  fácilmente.

Consideraremos concatenaciones  $w'_i w$ .

2. El conjunto  $\{w'_i w : w \in \text{Directos}_n, 1 \leq i \leq m(w)\}$  se ordena lexicográficamente. Llámese  $L_n$  a la lista resultante y denótese el renglón  $r$  de  $L_n$  por  $\iota_r \tau_r$  donde  $\iota_r$  y  $\tau_r$  consisten de  $n$  caracteres. La Tabla 1 muestra un ejemplo.

3. De  $L_n$  constrúyanse dos archivos,  $\text{Indices}_n$  y  $\text{Anagramas}_n$ , como sigue. El renglón inicial ( $r = 0$ ) de  $\text{Indices}_n$  es  $\iota_0 0$ , la concatenación de  $\iota_0$  con el entero 0. Si el renglón  $k$  de  $\text{Indices}_n$  se ha definido como  $\iota_r r$ , la concatenación de la cadena  $\iota_r$  con el entero  $r$ , definimos el renglón  $(k + 1)$  de  $\text{Indices}_n$  como  $\iota_s s$  donde  $s$  es el mínimo entero mayor que  $r$  tal que  $\iota_s \neq \iota_{s-1}$ . Si tal  $s$  no existe, no hay renglón  $(k + 1)$  en  $\text{Indices}_n$ . Para toda  $r$  el renglón  $r$  de  $\text{Anagramas}_n$  es  $\tau_r$ .

**Tabla 1.** Ejemplo que usa  $\text{Directos}_3 = \text{ARE, ERA, ERE}$ .

Renglón ( $r$ )	$w'_i$ (casi $\text{Indices}_3$ )	$w$ ( $\text{Anagramas}_3$ )
0	AER	ARE
1	AER	ERA
2	AE#	ARE
3	AE#	ERA
4	AR#	ARE
5	AR#	ERA
6	A##	ARE
7	A##	ERA
8	EER	ERE
9	EE#	ERE
10	ER#	ARE
11	ER#	ERA
12	ER#	ERE
13	E##	ARE
14	E##	ERA
15	E##	ERE
16	R##	ARE
17	R##	ERA
18	R##	ERE

En la Tabla 1 se puede ver un ejemplo, para un lexicón de 3 palabras, de la construcción de  $\text{Indices}_3$  y  $\text{Anagramas}_3$ , justo después de la fase 2. Las cadenas pequeñas de la Tabla 1 son repetitivas y desaparecen en la fase 3.

### 3.2. El generador de jugadas (el método de anagramas)

Describimos aquí el método de Heuri para generar todas las jugadas posibles.

La eficiencia del novedoso generador de jugadas se debe principalmente al uso adecuado de anagramas. Este generador es muy diferente de los utilizados por los motores más conocidos en la literatura sobre Scrabble; la mayoría usa DAWG o GADDAG para describir palabras y entonces seguir una trayectoria en la gráfica para verificar la existencia de palabras válidas.

En vez de ello, el generador de Heuri se aprovecha de la fuerza de los Anagramas. Las dos ideas claves son:

1) El almacenamiento de palabras en dos grupos de archivos ( $Indices_n$  y  $Anagramas_n$ ), donde  $n$  es el número de caracteres (incluyendo #) de las cadenas de  $Indices_n$ , y es también el número de letras de las palabras  $Anagramas_n$  ( $2 \leq n \leq 15$ ).

2) El enlace conveniente entre  $Indices_n$  y  $Anagramas_n$  ( $Indices_n$  es el grupo de cadenas de longitud  $n$  que tienen al menos un anagrama;  $Anagramas_n$  es el grupo de diferentes anagramas de longitud  $n$ ). Este enlace ayuda a viajar por el lexicon de la computadora para generar jugadas legales.

Defínase *segmento* como un conjunto de, al menos dos, cuadros consecutivos contenidos en un renglón o columna.

Los atriles y subatrilas son multiconjuntos de los caracteres de  $\{A, B, \dots, Z, \#\}$ , los cuales, a su vez, pueden identificarse con cadenas escritas lexicográficamente. El símbolo  $\oplus$  denota multisuma (unión con multiplicidades; ver Knuth p.694 [5]).

Para explicar el método de anagramas, además de la descripción esencial que damos a continuación, se muestra en el Algoritmo 1, un pseudocódigo. Este código genera todas las jugadas horizontales; un código análogo se usa para las jugadas verticales.

Supongamos dados un tablero jugado  $T$  y un atril. Queremos coleccionar todas las jugadas válidas.

Primeramente, usando  $T$ , se calculan el halo y los conjuntos  $H_\sigma$  y  $V_\sigma$  para cualquier cuadro  $\sigma$  del tablero. Ver Algoritmo 1

Después buscamos intervalos en un renglón dado usando  $T$ , el halo, los conjuntos  $H_\sigma$  y  $V_\sigma$  y el tamaño del atril dado. Cuando se encuentra un intervalo  $I$  se procede de la siguiente manera:

Sea *string* el multiconjunto formado por las letras de  $I \cap T$ . Sea  $I\_Rack$  la multisuma de *string* con las letras del atril. Sean  $I\_Subracks$  los multisubconjuntos de  $I\_Rack$ . Para cada  $I\_Subrack$  de  $I\_Subracks$  obténganse todos los anagramas y trátense de colocar en el intervalo  $I$  (véase el Algoritmo 1).

El  $I\_Subrack$  se ordena lexicográficamente para encontrar todos sus anagramas, usando tablas hash donde los  $Indices_n$  son las claves y  $Anagramas_n$  son los valores. Entonces se trata de colocar cada anagrama en el intervalo usando las coordenadas iniciales de  $I$  y los conjuntos  $H_\sigma$  y  $V_\sigma$ . Finalmente, si el anagrama puede colocarse, una jugada válida se ha encontrado, la cual se guarda (ver la última parte del Algoritmo 1).

---

**Algoritmo 1** El Método de Anagramas

---

```

1: halo=CalcularHalo( $T=$ TableroJugado)
2:  $(H, V)=$ CalcularConjuntosH-y-V( $T=$ TableroJugado, Atril,  $Directos_n, Inversos_n$ )
3: // Generación de jugadas horizontales
4: for renglón = 1 to 15 do
5:   for For icolumna=1 to 14 do
6:     for For tcolumna= icolumna+1 to 15 do
7:       segmento= [ (renglón, icolumna), (renglón, tcolumna) ]
8:       if (segmento  $\cap$  halo)  $\neq \emptyset$  then
9:         if (renglón, icolumna-1)  $\notin T$  and (renglón, tcolumna+1)  $\notin T$  then
10:          if |segmento- $T$ |  $\leq$  TamañoAtril then
11:            // Se encontró un Intervalo
12:            string = multiconjunto de letras en el segmento
13:             $I\_Rack = Rack \uplus string$ 
14:             $n=tcolumna-icolumna+1$ 
15:             $I\_Subracks =$  GeneralRackSubsets.deLongitud_n.Dada( $I\_Rack, n$ )
16:            for all  $I\_Subrack$  in  $I\_Subracks$  do
17:               $I\_Subrack =$  OrdenaLex( $I\_Subrack$ )
18:              Anagramas= EncuentraAnagramas( $I\_Subrack, Indices_n, Anagramas_n$ )
19:              // Hay jugadas válidas si Anagramas encajan en Intervalos
20:              for all Anagramas do
21:                if PonAnagramas(Anagrama, renglón, icolumna,  $H, V$ ) then
22:                  JugadaVálida=(renglón, icolumna, Anagrama)
23:                end if
24:              end for
25:            end for
26:          end if
27:        end if
28:      end if
29:    end for
30:  end for
31: end for

```

---

## 4. Conclusiones

Para mostrar la velocidad del generador de jugadas de Heuri usando el método de anagramas lo comparamos con el de Quackle 2015 jugando 1000



partidas Quackle vs. Heuri, en inglés, usando el lexicón twl06 y usando Quackle GADDAG para almacenar el lexicón; se obtuvieron los siguientes resultados:

El promedio de turnos fue 24.25 . El número máximo de turnos fue 37 . El promedio de jugadas legales analizadas por juego fue 23541.95 . El factor de ramificación ( promedio de jugadas legales por turno ) fue 970.88 . Tiempo promedio de Quackle por juego: 331.26 ms. Tiempo promedio de Heuri por juego: 126.69 ms. Razón de tiempos Quackle/Heuri (Q/H) por juego: 2.61 . Razón de tiempos Q/H en las primeras 12 jugadas: 5.14 .

Las comparaciones de tiempos por jugada entre Quackle y Heuri, después de 1000 partidas, se muestran en la Tabla 2.

**Tabla 2.** Comparaciones de tiempos por jugada entre Quackle y Heuri.

N. de Jugada	T. de Quackle	T. de Heuri	Razón de Tiempos Q/H
1	5.93 ms.	0.3 ms.	19.58
2	9.59 ms.	0.95 ms.	10.07
3	13.14 ms.	1.46 ms.	8.99
4	13.71 ms.	1.87 ms.	7.33
5	14.49 ms.	2.28 ms.	6.37
6	13.03 ms.	2.64 ms.	4.93
7	14.93 ms.	3.13 ms.	4.77
8	17.08 ms.	3.65 ms.	4.68
9	15.7 ms.	4.02 ms.	3.91
10	17.15 ms.	4.59 ms.	3.74
11	14.45 ms.	4.91 ms.	2.94
12	19.43 ms.	5.61 ms.	3.46
13	13.97 ms.	5.81 ms.	2.4
14	16.55 ms.	6.42 ms.	2.58
15	15.63 ms.	6.82 ms.	2.29
16	16.67 ms.	7.47 ms.	2.23
17	15.18 ms.	7.66 ms.	1.98
18	15.16 ms.	8.42 ms.	1.8
19	13.14 ms.	8.47 ms.	1.55
20	13.83 ms.	9.14 ms.	1.51
21	9.02 ms.	7.83 ms.	1.15
22	8.33 ms.	7.3 ms.	1.14
23	4.06 ms.	4.53 ms.	0.9
24	3.23 ms.	3.33 ms.	0.97

En promedio al comparar el método de anagramas de Heuri con el método GADDAG usado por Quackle, el método de anagramas resulta ser 2.61 veces más rápido que el método GADDAG cuando se juega una partida completa.

La razón de tiempos Q/H disminuye cuando el número de jugadas aumenta (ver Tabla 2); siendo las 12 primeras jugadas en las que el método de anagramas de Heuri utiliza menos tiempo en comparación con el método GADDAG utilizado por Quackle. Las primeras 12 jugadas son más significativas que las siguientes;

esto se debe a una característica que se da en los juegos de Scrabble: a menudo un jugador obtiene ventaja en las primeras jugadas y es difícil para el contrincante recuperarse de esta desventaja. Después de jugar 1000 partidas, el 75 % de las veces la partida fue ganada por quien iba ganando en la jugada 12.

En las primeras 12 jugadas la razón de tiempos Q/H es 5.14. Por tanto, el método de anagramas permite más tiempo que el GADDAG para analizar las 12 primeras jugadas lo cual, por lo explicado anteriormente, podría resultar en un mejor desempeño.

Para ilustrar las comparaciones de tiempo entre los generadores de jugadas cuando aparecen comodines ( # ), se hicieron tres experimentos usando el lexicon en inglés con una sola palabra en el tablero ( 8F ANEROID ). Los experimentos consistían en producir todas las jugadas posibles para 3 atriles diferentes dados. En el primer experimento el atril tenía 2 comodines ( # ); en este experimento Heuri resultó ser 31 veces más rápido que Quackle (que usaba GADDAG ). En los otros dos experimentos el atril tenía un comodín en uno y cero en el otro; véase la Tabla 3. Estos experimentos ilustran como el método de anagramas es mucho más rápido que el método GADDDAG cuando aparecen comodines.

Un *callejón sin salida* es una trayectoria en un DAWG que no es una palabra válida, pero es el principio de una palabra válida. Por ejemplo, el camino  $c \rightarrow a \rightarrow r \rightarrow a \rightarrow m \rightarrow b$  denota *caramb* que no es una palabra válida, pero es el principio de una palabra válida.

Los métodos GADDAG y DAWG tienen muchos callejones sin salida, DAWG más que GADDAG. Estos cuestan tiempo en la generación de jugadas. Una ventaja del método de anagramas es que NO hay callejones sin salida, esta cualidad lo ayuda a ser más rápido que los otros dos métodos. Además, es mucho más rápido que el GADDAG cuando aparecen comodines porque, en tal caso, el número de callejones sin salida aumenta considerablemente. Véase la Tabla 3.

**Tabla 3.** Tiempos para generar todas las jugadas con un tablero no vacío.

Atril	Tiempo de Quackle	Tiempo de Heuri	Razón de Tiempos Q/H
<b>AENRS##</b>	<b>1431.1 ms.</b>	<b>45.7 ms.</b>	<b>31.3</b>
AEINRS#	297.2 ms.	10.9 ms.	27.3
AEINRST	25.3 ms.	2.1 ms.	12

El uso de anagramas, tablas hash, intervalos junto con las restricciones impuestas por las  $H_\sigma$  y  $V_\sigma$ , hacen muy rápido al generador de jugadas de Heuri (el Método de Anagramas). La cantidad de memoria RAM que se necesita es entre 300 MB y 400 MB (dependiendo del lexicon utilizado); es una cantidad razonable para las computadoras actuales.

El método de anagramas se usará mucho más al incorporar la simulación en Heuri, para propósitos de mejorar su defensa. Entonces se volverán a confrontar los motores de Heuri y Quackle (esta vez ambos contarán con simulación).

Esperamos que la velocidad del método de anagramas sea muy útil para ayudar a Heuri a lograr un buen resultado contra Quackle.

## Referencias

1. Appel, A.W., Jacobson, G.J.: The World's Fastest Scrabble Program. *Communications of the ACM*, 31(5), pp. 572-578 (1988)
2. González-Romero, A., Alquézar, R., Ramírez-Flores A., González-Acuña, F.: Heuristics and Fishing in Scrabble. *Chairs : Tristan Cazenave, Jean Mehat, Mark Winands (Eds.), In: Proceedings of the European Conference on Artificial Intelligence, (ECAI'12), Computer Games Workshop (CGW12) (2012)*
3. Gordon-Steven, A.A.: Faster Scrabble Move Generation Algorithm. *SoftwarePractice and Experience*, 24(2), pp. 219-232 (1994)
4. Katz-Brown, J., O'Laughlin, J., Fultz, J., Liberty, M.: Quackle is an open source crossword game program (2006)
5. Knuth, D.E.: *The Art of Computer Programming. 2, Seminumerical Algorithms* (1997)
6. Ramírez, A., González-Acuña, F., González-Romero, A., Alquézar, R., Hernández, E., Roldán-Aguilar, A., García-Olmedo, I.: A Scrabble Heuristic Based on Probability That Performs at Championship Level. In: *Proceedings of the 8th Mexican International Conference on Artificial Intelligence MICAI*, pp. 112-123 (2009)
7. Scrabble letter distributions: <https://en.wikipedia.org/wiki/Scrabble> (2012) `_letter_distributions`
8. Scrabble pages: <http://www.scrabblepages.com/scrabble/rules/> (2012)
9. Scrabble wikipedia: <https://en.wikipedia.org/wiki/Scrabble> (2012)



## Identificación del perfil de usuario en Twitter utilizando recursos semánticos

J. Víctor Carrera-Trejo, Miguel Á. Álvarez-Carmona, Luis Villaseñor-Pineda

Instituto Nacional de Astrofísica, Óptica y Electrónica,  
Laboratorio de Tecnologías del Lenguaje,  
México

jvcarrera@{ipn, inaoep}.mx, {miguelangel.alvarezcarmona, villasen}@inaoep.mx

**Resumen.** Los *hashtags*, las menciones de usuario o las direcciones *url* compartidas en Twitter son características de esta red social que pueden ser útiles para observar los intereses de un usuario. El presente trabajo evalúa la posibilidad de usar este tipo de características para identificar el perfil del usuario. No obstante, dada la variabilidad y especificidad de dichas características no es posible usarlas directamente, por lo que es necesario determinar el concepto asociado a través de recursos semánticos. En el caso particular del trabajo mostrado en este artículo se comprobó la utilidad del grafo de conocimiento de *Google*. Dicho grafo se construye a partir de los documentos públicos en Internet, reuniendo y asociando todo tipo de información de manera dinámica. La evaluación del método propuesto se realizó usando el corpus en inglés del PAN 2014. Los resultados alcanzados evidencian que la información de estas características puede aprovecharse en el perfilado de autores.

**Palabras clave:** clasificación no temática, perfilado de autor, Google, Twitter, grafo de conocimiento, hashtags, PAN 2014.

### Author Profiling in Twitter using Semantic Resources

**Abstract.** Hashtags, user mentions or url addresses shared on Twitter are features of this social network that can be useful to observe the interests of a particular user. The present work evaluates the possibility of using this kind of characteristics to identify the user's profile. However, given the variability and specificity of these characteristics, it is not possible to use them directly, so it is necessary to determine the associated concept or meaning through semantic resources. In the particular case of the work presented in this article, the use of the Google's knowledge graph was verified. This kind of graph is built using public documents found on the internet, gathering and associating dynamically all kind of information. The evaluation of the proposed method was carried out

using the english corpus of the PAN 2014. The results obtained show that the information of these characteristics can be used in the author profiling.

**Keywords:** Twitter, knowledge graph, hashtags, classification, Google, author profiling, PAN 2014, concepts.

## 1. Introducción

El uso de internet ha generado nuevas formas de comunicarse, en las cuales se hace uso de diferentes aplicaciones para compartir información, entre estas aplicaciones se pueden encontrar blogs, microblogs, foros, etc. En ellas el principal objetivo es compartir información relacionada con diferentes temáticas o tópicos, ejemplos de estos son noticias, opiniones, revisiones de productos o servicios, etc. En algunas ocasiones la información se comparte de forma anónima, por lo que se desconocen datos acerca de quien realiza la publicación. Datos que podrían ser de interés por distintas razones; por ejemplo, para mercadotecnia o seguridad.

Dentro de las diferentes áreas de investigación de procesamiento del lenguaje natural existen diferentes tareas enfocadas a descubrir información relacionada con el autor de un texto [1], la primera de ellas, denominada “*author profiling*” o perfil de autor, se enfoca en identificar rasgos del autor de un documento como su edad, su género, su profesión, entre otros. Otra tarea se denomina “*author identification*” o identificación de autor, en la cual se intenta identificar al autor de un texto anónimo de entre un conjunto de autores dado.

Como se puede intuir, ambas tareas se pueden resolver con un enfoque de clasificación supervisado extrayendo características de textos con autores conocidos y a partir del análisis de estas características resolver la tarea que se requiera.

Existen diferentes tipos de características que pueden ser extraídas de los textos utilizados. Dentro de estas características [2,23] se pueden encontrar combinaciones léxicas, aquellas basadas en el estilo, como pueden ser signos de puntuación, el uso de mayúsculas, la longitud de las frases, etc., características semánticas [3,4] obtenidas mediante el uso de algoritmos de semántica latente o aquellas extraídas a partir del uso de etiquetadores de partes de la oración (*part-of-speech taggers*, POS), que permiten conocer la categoría gramatical a la que pertenece una palabra dentro de un texto [24].

En principio, la tarea de “*author profiling*” se aplicó a textos formales, como noticias o libros; sin embargo, en los últimos años se ha tratado de determinar características de usuarios en redes sociales a través de los textos que ellos mismos comparten.

Una de las plataformas más utilizadas y estudiadas actualmente es Twitter, la cual es una red de servicio de microblogging que cuenta con más de 330 millones de usuarios, los cuales publican más de 500 millones de mensajes diariamente [7], también conocidos como tweets, los cuales tienen una longitud limitada de caracteres, en los que se pueden incluir enlaces (*url*) a sitios webs externos, imágenes o videos que puedan ser vistos por otros usuarios que tengan acceso al microblog. Esta red social ofrece una gran fuente de información y ha sido motivo de muy diversas investigaciones, entre las que se pueden encontrar: minería de opinión, análisis de

sentimientos, predicción de resultados electorales, estudios de mercado, análisis de desastres, etcétera.

Esta red social presenta un tipo propio de características, conocidas como interactivas [6], ya que éstas ofrecen un medio de interacción con otros usuarios, compartiendo menciones de otros usuarios, contenido mediante direcciones url o hashtags. Haciendo uso de los hashtags los usuarios comparten información temática, la cual es representada por el texto que el mismo usuario define. Sin embargo, muchas veces la interpretación del hashtag depende del contexto, del usuario o del dominio en que se comparte. De ahí que a pesar de ser “etiquetas” sobre un tema o un concepto, se necesite de un recurso para su interpretación. Por este motivo es necesario contar con alguna herramienta o recurso semántico a partir del cual extraer la información que un hashtag representa. Una herramienta que contiene dicha información es el grafo de conocimiento de Google [8,9]. Este grafo brinda información en general y puede ser usado para recuperar el concepto descrito por un hashtag.

El objetivo de este trabajo es analizar si el uso de características propias de Twitter, principalmente las url y hashtags compartidos, en conjunto con el grafo de conocimiento de Google [8,9], son útiles en la tarea de perfil de autor.

El resto de este artículo se estructura de la siguiente manera: En la sección 2 se presenta el trabajo relacionado; la sección 3 describe el corpus y el método propuesto; en la sección 4 se muestran los resultados obtenidos y, finalmente, en la sección 5 se analizan los resultados ofreciendo algunas conclusiones y comentarios acerca de posibles trabajos futuros.

## **2. Trabajo relacionado**

Antes de abordar la descripción de los trabajos relacionados se describen algunos conceptos centrales para el trabajo propuesto.

### **2.1. Caracterización de la información**

Existen diferentes formas de representación de los textos en tareas de clasificación, siendo la más común de ellas el modelo espacio vectorial [13,14,15], en lo cual el documento es representado como un vector, donde los valores de los componentes del vector representan los valores de las características extraídas del documento, por lo que el tamaño del vector corresponde con el número total de características. Estas características son usualmente palabras o algunas de sus invariantes morfológicas, como pueden ser sus lemas, entre otras.

El valor de cada característica se calcula de acuerdo con un tipo de pesado en particular [2,16], entre los que se encuentran principalmente binario, *tf*, *idf* y su combinación. La forma de caracterización más simple es la caracterización binaria, en la cual se debe indicar mediante un valor de “1” si una palabra en particular del vector de características se encuentra dentro de un texto en particular a caracterizar o bien se indica mediante un valor de “0” cuando dicha palabra no se encuentra.

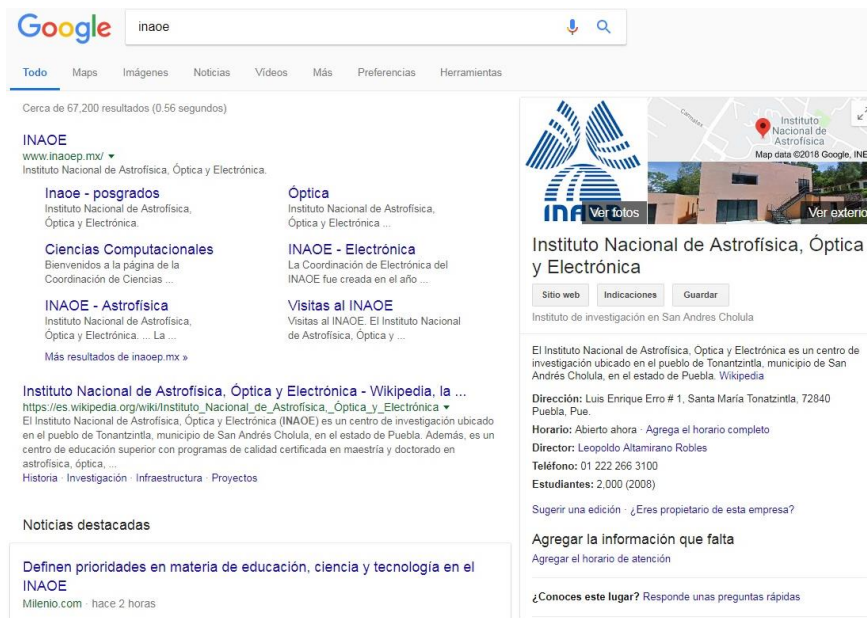


Fig. 1. Grafo de conocimiento para el término “INAOE”.

## 2.2. Clasificación de la información

La clasificación de documentos busca determinar si un documento pertenece a una o varias categorías, también conocidas como clases o etiquetas. La clasificación de textos está basada en técnicas de *machine learning* [17,18], entre estas técnicas se pueden mencionar los clasificadores Naïve Bayes, máquinas de soporte vectorial con diferentes kernels, árboles de decisión, redes neuronales, etcétera.

## 2.3. El grafo de conocimiento de Google

El grafo de conocimiento de Google [8] es una base de conocimiento creada y utilizada por los servicios de Google. Para su creación se utilizan diferentes fuentes de información; por ejemplo, Freebase, Wikipedia, CIA World Facebook, entre otras. El grafo cuenta actualmente con más de 500 millones de objetos y 3.5 billones de hechos y relaciones entre estos objetos, y gracias a este grafo es posible obtener información relacionada con personas, eventos, lugares etcétera.

Esta herramienta puede ser consultada utilizando el motor de búsqueda de Google de dos formas, utilizando un web browser o un API de programación. En la figura 1 se observa la consulta de la palabra “INAOE”, donde la información devuelta por el grafo de conocimiento se encuentra dentro del rectángulo del lado izquierdo. En dicho rectángulo se muestra la información relacionada con el concepto principal al que se asocia el término buscado; por ejemplo, de la consulta realizada, el término “INAOE”



se asocia con un centro de investigación del cual se puede conocer su dirección, horario, nombre de su director, entre otras. Por otro lado, al realizar la consulta utilizando un API de programación [10] es posible obtener el conjunto de conceptos asociados al término, como son, entre otros, una descripción del objeto y su tipo.

Por ejemplo, al realizar consulta de la palabra “INAOE” utilizando el API de Google, se obtienen, entre otros, los siguientes términos relacionados:

1. “name”: “14674 INAOE”, “description”: “Asteroid”, “@type”: [“Thing”].
2. “name”: “National Institute of Astrophysics, Optics and Electronics”, “description”: “Research institution in San Andres Cholula, Mexico”, “@type”: [“Thing, “Organization, Place, EducationalOrganization, CollegeOrUniversity”].
3. “name”: “Héctor Manuel Moya Cessa”, “description”: “Physicist”, “@type”: [“Person, Thing”].
4. “name”: “Guillermo Haro”, “description”: “Mexican astronomer”, “@type”: [“Person, Thing”].
5. “name”: “Atacama Cosmology Telescope”, “description”: “Observatory in Chile”, “@type”: [“Place, Thing”].

## 2.4. Trabajos relacionados

Diversos trabajos se enfocan en buscar el mejor conjunto de características que permitan resolver la tarea de *autor profiling*; por ejemplo, en [12] se presentan diferentes trabajos que hacen uso de diferentes tipos de características: de estilo (signos de puntuación, tamaños de las sentencias, número caracteres, etc.), léxicas (*n*-gramas o bolsa de palabras), temáticas (utilizando recursos como LIWC) o bien, características basadas en representaciones distribucionales identificando relaciones entre términos, documentos, perfiles y sub-perfiles [25].

En otros trabajos, como en [6], los autores se enfocan a analizar características que surgen a partir de los textos de twitter, denominándolas “*social behavioral biometrics*”, como son los hashtags, las menciones a usuario o los url compartidos para así poder inferir datos de los usuarios como patrones de comportamiento, de comunicación, entre otros. En [19] los autores realizan la clasificación temática de los hashtags de un corpus de twitter, utilizando una máquina de soporte vectorial como clasificador, utilizando un etiquetador temático y Wikipedia.

Finalmente, en [9] se presenta el concepto de baúl del conocimiento (*knowledge vault, KV*), el cual se refiere a construir un gran repositorio de información a partir de la consulta online de diferentes bases de conocimiento estructuradas, como son, entre otras, Wikipedia, Freebase, YAGO, Satori de Microsoft, incluyendo el grafo de conocimiento de Google. En el KV se almacena información adicional de un concepto a partir de su búsqueda en las diferentes bases de conocimiento, esta información es almacenada en forma de una tripleta (sujeto, predicado, objeto). Por ejemplo, para los términos “Barack Obama”, el sistema almacena la tripleta (“Barack Obama”, “Place of birth”, “Honolulu”), entre muchas otras que describen el concepto “Barack Obama”.

Como se observa en párrafos anteriores, existen diferentes trabajos que proponen el uso de diversos tipos de características para identificar los temas de interés para un autor y con ellos, en conjunto de información estilística, intuir el perfil de dicho autor.

En el caso particular de Twitter, los hashtags son etiquetas propuestas por los usuarios para nombrar un tema. Desafortunadamente, el hashtag no puede interpretarse directamente, de ahí que para obtener las características temáticas sea necesario el uso

**Tabla 1.** Conceptos extraídos a partir del grafo de conocimiento.

Hashtag	Conceptos
#facebook	video_sharing_company technology_company social_network_company drama_series book_by_david_kirkpatrick song_by_rhett_and_link broadcasting_television_network
#graffiti	visual_art_type 1973_film studio_album_by_chris_brown studio_album_by_led_zeppelin video_game_series 1979_film 1990_film american_singer-songwriter comic_magazine_series company
#beirut	capital_of_lebanon, band, university_in_beirut_lebanon, country_in_the_middle_east city_in_oregon
#shazam	fictional_superhero television_program fictional_character 2019_film company rock_band american_television_show comic_series studio_album_by_the_move
#concacaf	league football_competition tournament competition sports_association soccer_team
#japan	country_in_east_asia soccer_team music_company state capital_of_japan war internet_services_company
#youtube	video_sharing_company orchestra swedish_comedian award_ceremony television_company event court_case american_sitcom public_university_in_milton_keynes_england

recursos adicionales. El presente trabajo analiza el uso de un recurso semántico: el grafo de conocimiento de Google, para evidenciar el o los conceptos detrás de un hashtag. A través de esta transformación se espera impactar el proceso de clasificación en la tarea de perfil de autor.

### 3. Metodología

Para este trabajo se utilizó el corpus del PAN-2014, el cual es descrito en la sección 3.1. En los apartados subsecuentes se describen los pasos utilizados en la metodología presentada en este trabajo: en la sección 3.2 se describe el proceso de tokenización y la obtención de los conceptos a partir de la consulta del grafo de conocimiento de Google; finalmente, en la sección 3.3 se indica el proceso para construir los diferentes conjuntos de características mencionados en la sección 3.2 y su clasificación, calculando los valores de las medidas precisión, cobertura (o recall) y *f1* para su posterior análisis y comparación.

#### 3.1. Corpus PAN 2014

Una de las tareas del PAN-2014 [11,12] es la de *author profiling*, en la que el objetivo es: dado un documento identificar el género y la edad de su autor. Para llevar a cabo la tarea se construyó un corpus que incluye textos de blogs, revisiones de hoteles y twitter, en dos idiomas: inglés y español. Con base en este corpus se extrajo la parte relacionada con twitter en idioma inglés, y se obtuvieron 306 archivos.

En el caso de género, el corpus se encuentra balanceado ya que el corpus cuenta con 153 archivos para género femenino y 153 para género masculino. Sin embargo, en el

Tabla 2. Características.

Característica	Descripción
normales	Tokens extraídos del corpus, en minúsculas, se incluyen las stopwords.
lematizadas	Tokens que pertenecen a la característica “normales”, se lematizan todos ellos excepto las características de twitter.
urls	Urls extraídas del corpus.
dominios	Dominios de las urls extraídas del corpus, se sustituyen cada una de las urls por estos dominios.
hashtags	Hashtags extraídos del corpus.
grafo	Conceptos indicados por el grafo del conocimiento para los hashtags, se sustituyen cada uno de los hashtags por estos conceptos.
usuarios	Usuarios mencionados en los tweets.

caso de edad el corpus no se encuentra balanceado, ya que se tienen 5 clases con diferentes números de archivos: se tiene 20 archivos que corresponden a autores cuyo rango de edad se encuentra entre 18 y 24 años; 88 corresponden a aquellos entre 25 y 34; 130 se relacionan a personas entre 35 y 49; 60 a aquellos entre 50 y 64 y, finalmente, solamente 8 corresponden a autores con más de 65 años.

### 3.2. Caracterización del corpus y grafo del conocimiento

Para realizar la caracterización del corpus se tomó cada uno de los 306 archivos y utilizando la herramienta de software *Ark Tokenizer* [5], se separó cada archivo en tokens, incluyendo cada uno de ellos en cada uno de los 5 conjuntos de características definido por la herramienta.

Se lematizaron cada uno de los tokens obtenidos, excepto aquellos que pertenecen al conjunto denominado “*twitter/online specific*” ya que contiene características específicas de twitter, utilizando para ello la herramienta *CoreNLP* de Stanford [20], obteniendo así conjuntos de características lematizadas y normales.

Utilizando el conjunto de características de twitter se separaron las direcciones url y a partir de cada una de ellas se separó el dominio de la url; por ejemplo, de la siguiente url compartida “*http://bit.ly/j5I178*” el dominio extraído es “*bit.ly*”, con esto se obtienen dos conjuntos, uno de ellos normales y el otro denominado dominios.

Para extraer la información relacionada con cada uno de los hashtags aplicando el grafo del conocimiento, se desarrolló una aplicación en el lenguaje de programación Python [21], donde se realizó la consulta de cada hashtag en el grafo. De la información devuelta para cada consulta se extrajeron los datos mostrados en el campo “*description*”. Algunos conceptos extraídos se muestran en la tabla 1, en esta tabla se observa, por ejemplo, que para el hashtag #facebook, el grafo de conocimiento de Google lo identifica como una red social o una compañía en la que se comparten videos, aunque también lo identifica en menor medida como una canción o un libro, otro ejemplo son los conceptos relacionados con #japan, éste es relacionado con un país o un equipo de soccer.

Finalizado el proceso de caracterización del corpus, se construyeron los diferentes conjuntos de características, las cuales se describen en la tabla 2.

### 3.3. Construcción y clasificación de los conjuntos de características

Considerando las características mostradas en la tabla 2 y la descripción del corpus que se realizó en la sección 0, se construyeron los diferentes archivos para la tarea de clasificación. En el caso de la clasificación para género se consideró un esquema *10-fold cross validation*, construyendo un archivo de entrenamiento y uno de validación por *fold*. Por otro lado, para el caso de edad se utilizó un esquema *5-fold cross validation*, ya que en este caso las 5 clases no se encuentra balanceadas y se buscó incluir principalmente archivos de todas las clases tanto en el archivo de entrenamiento como en el de prueba.

Por otro lado, los vectores de caracterización de cada uno de los archivos se construyeron utilizando un esquema de pesado binario y considerando los tokens que se encuentren en dos o más archivos.

Finalmente, utilizando la librería *Scikit-Learn* [22] de Python [21] se realizó la clasificación de los diferentes *folds*. Para medir la eficacia de la representación propuesta se aplicaron las medidas *precision* y *recall* para por clase, así como los promedios generales. Para el cálculo de la medida F1 se utilizó la ecuación mostrada en (1), tanto por clase, como de forma general:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (1)$$

## 4. Experimentos

En esta sección se mostrarán los resultados obtenidos por la representación propuesta en este trabajo. Como clasificador se utilizó el algoritmo SVM de Scikit-Learn [22].

### 4.1. Bolsa de palabras

Para la bolsa de palabras, en el caso del género se obtuvieron los resultados que se muestran en la tabla 3. Como se puede observar, el mejor resultado se obtuvo al utilizar el conjunto de los lemas de las palabras del corpus y sustituyendo las url compartidas por los dominios de ellas, que se denomina *lematizadas\_dominios*. En segundo término, se encuentran los tokens del conjunto de características normales, las url que se incluyen dentro de este conjunto de características que se encuentran como en el corpus, este conjunto es llamados normales\_urls.

A continuación, se encuentra el conjunto de características normales donde se utilizaron los dominios de las url en lugar de todas ellas, dicho conjunto se identifica como *normales\_dominios* y, finalmente, el conjunto llamado *lematizadas\_urls* en el que se incluyen los lemas de los tokens y donde las url compartidas no sufren ninguna modificación, una descripción más detallada del conjunto de características se encuentra en la tabla 2. Finalmente, cabe hacer notar que los valores de F1 no varían demasiado entre los diferentes conjuntos.

**Tabla 3.** Resultados macro de clasificación para género.

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
lematizadas_dominios	0.81075	0.8	0.79745
normales_urls	0.7985	0.78333	0.78003
normales_dominios	0.80205	0.78333	0.77955
lematizadas_urls	0.78832	0.78	0.77772

**Tabla 4.** Resultados macro de clasificación para edad.

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
lematizadas_dominios	0.39064	0.45	0.40389
normales_urls	0.36542	0.44333	0.3928
lematizadas_urls	0.37096	0.43667	0.39162
normales_dominios	0.35124	0.43	0.38097

En el caso de la clasificación por edad, los resultados se muestran en la tabla 4, como se puede observar al igual que en el caso de género el conjunto de características mejor clasificado es el conocido como *lematizadas\_dominios*.

De acuerdo con los valores obtenidos para género y edad, el conjunto denominado como *lematizadas\_dominios* obtuvo los valores más altos de F1, por lo que estos serán utilizados como *baseline* en los experimentos posteriores.

#### 4.2. Características de twitter extraídas del corpus

Utilizando las diferentes características propias de twitter se tienen los siguientes resultados de clasificación. Para el caso de género, estos se muestran en la tabla 5. Como se puede observar, el conjunto de características que obtiene el mejor valor de F1 son los dominios que se extraen de las url compartidas. Un valor similar, pero por debajo, es el que se obtiene al utilizar las menciones de usuarios, los hashtags y sus conceptos extraídos a partir del grafo. El peor resultado es el que se indica al utilizar las url sin modificarlas. Es importante mencionar que ninguno de estos resultados mejora el *baseline* propuesto.

Para el caso de la clasificación por edad, se obtuvieron los siguientes datos mostrados en la tabla 6. Se puede ver que nuevamente el uso de los dominios ofrece el mejor resultado de F1; sin embargo, los conceptos representados por el grafo obtienen un resultado muy similar al del *baseline*. Sin embargo, los resultados obtenidos por los hashtags, las url completas y las menciones de usuarios se encuentran por debajo de éste.

Finalmente, en las tablas 7 y 8 se pueden revisar los valores obtenidos de la medida F1 micro para cada una de las clases para género y edad, respectivamente, y las características utilizadas.

Para género, el mejor resultado para ambos sexos se obtiene utilizando el *baseline*, analizando las características de twitter, el uso de dominios obtiene buenos resultados tanto para el género femenino y masculino, aunque las menciones de usuarios obtienen el mejor resultado para el género femenino.

**Tabla 5.** Resultados macro para la clasificación de género.

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
dominios	0.65509	0.64	0.63115
usuarios	0.63388	0.61667	0.60237
hashtags	0.61169	0.6	0.59026
grafo	0.56705	0.564	0.55648
urls	0.58497	0.52333	0.45354

**Tabla 6.** Resultados macro para la clasificación de edad.

	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
dominios	0.41255	0.43333	0.41746
grafo	0.41038	0.42745	0.41143
hashtags	0.35619	0.36667	0.34743
urls	0.42057	0.42667	0.32523
usuarios	0.31029	0.32667	0.30802

**Tabla 7.** F1 micro para género.

<b>Características</b>	<b>Femenino</b>	<b>Masculino</b>
lematizadas_dominios ( <i>baseline</i> )	0.809	0.787
grafo	0.562	0.551
hashtags	0.643	0.536
usuarios	0.674	0.53
urls	0.642	0.263
dominios	0.666	0.598

**Tabla 8.** F1 micro para edad.

<b>Característica</b>	<b>18-24</b>	<b>25-34</b>	<b>35-49</b>	<b>50-64</b>	<b>65+</b>
lematizadas_dominios ( <i>baseline</i> )	0.0	0.438	0.572	0.162	0.0
grafo	0.18	0.4	0.524	0.256	0.0
hashtags	0.124	0.29	0.48	0.242	0.0
usuarios	0.058	0.262	0.444	0.19	0.0
urls	0.0	0.162	0.582	0.134	0.0
dominios	0.168	0.404	0.518	0.336	0.0

Para edad, se puede observar que el uso de los dominios o conceptos del grafo de conocimiento como característica ofrecen los mejores resultados en 4 de 5 clases, en comparación con el *baseline*, esta última característica ofrece buenos resultados en 3 de 5 clases, siendo la clase de 65 o más años donde ninguna característica obtiene resultados al clasificarse, pero en la 18 a 24 años el *baseline* no obtiene resultados, pero el uso de conceptos o dominios si los obtienen.

## 5. Conclusiones y trabajo futuro

A partir de los resultados obtenidos para la tarea de perfil de autor se puede observar que, para el caso de género, el conjunto de características que ofrece los mejores

resultados son aquellas que se basan en la utilización de todas las palabras, siendo el conjunto que se compone por los lemas de los tokens y los dominios de las url el que mejor resultado obtiene. Cabe hacer notar que no presenta una gran diferencia con los otros conjuntos de características basadas en bolsa de palabras. Por otro lado, con respecto a los resultados obtenidos utilizando las características propias de twitter no se observa ninguna mejora. Al analizar cada característica, en particular se observa que el uso de los dominios de las url mejora la clasificación en el caso del género masculino, lo que permite presuponer que los hombres comparten diferentes tipos de contenido pero que proviene de sitios similares. Mientras para el caso del género femenino el uso del dominio o de toda la url ofrece resultados similares, lo que puede interpretarse como que comparten información de sitios distintos. En el caso de las menciones de usuario, hashtags y conceptos identificados vía el grafo de conocimiento ofrecen resultados similares para el caso de los hombres, por lo que comparten contenidos similares y en el caso de las mujeres, la clasificación cae utilizando sólo los conceptos.

Para el rasgo de edad se observa que el uso de conceptos del grafo de conocimiento y el uso de dominios presenta una ligera mejora en la clasificación, de acuerdo con el conjunto de características basado en el uso de lemas y dominios, caso contrario para los hashtags, los usuarios y las url. A nivel micro se tiene que el uso de conceptos o dominios mejora la clasificación en edades de 35-49 y 50-64, no así en el resto de ellas, resultados que a su vez mejoran la clasificación final.

En general se identifica que el uso de conceptos extraídos a partir de hashtags y dominios de las url compartidas como características en la tarea de perfil de autor permite obtener resultados interesantes. Aunque es importante mencionar que esto depende de la presencia de hashtags en el corpus a analizar, por lo que como trabajo futuro es importante realizar un estudio estadístico de la cantidad de url y hashtags que contienen las clases que mejoraron.

Por otro lado, la identificación de conceptos utilizando herramientas semánticas como el grafo de conocimiento, permite conocer el significado de características temáticas, por lo que pueden ser útiles en otras tareas de minería de textos. Un trabajo a futuro es comparar los resultados que se obtienen con ellas con los obtenidas utilizando algoritmos de semántica latente.

Otra vertiente a explorar es la combinación con técnicas de análisis de sentimientos. Ya que una vez que se obtiene el tema asociado a un hashtag también es posible identificar la aprobación o rechazo a ese tema.

Información que enriquecería la caracterización del mensaje y ayudaría a una mejor identificación de los rasgos del perfil del usuario.

**Agradecimientos.** Este trabajo fue desarrollado con el apoyo parcial del CONACYT bajo el programa de posdoctorados nacionales.

## **Referencias**

1. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the reproducibility of PAN's shared tasks: Plagiarism detection, author identification, and author

- profiling. In: Information Access Evaluation. Multilinguality, Multimodality, and Interaction, pp. 268–299 (2014)
2. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Siordia, O.S., Villaseñor, E.A.: A case study of spanish text transformations for twitter sentiment analysis. In: Expert Systems with Applications, pp. 457–471. DOI: 10.1016/j.eswa.2017.03.071. 81 (2017)
3. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. In: Journal of the American society for information science, American Documentation Institute, 41(391) (1990)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: The Journal of Machine Learning Research, 3, pp. 993–1022 (2003)
5. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for twitter: Annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers, 2, pp. 42–47 (2011)
6. Sultana, M., Paul, P.P., Gavrilova, M.: Mining social behavioral biometrics in twitter. In: Proceeding (CW'14), Proceedings of the International Conference on Cyberworlds, IEEE Computer Society Washington (2014)
7. Statista Homepage: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/> (2014)
8. Google: <https://googleblog.blogspot.mx/2012/05/introducing-knowledge-graph-things-not.html> (2012)
9. Dong, X.L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmman, T., Sun S., Zhang, W.: Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In: The 20th (ACM SIGKDD) International Conference on Knowledge Discovery and Data Mining, (KDD'14), pp. 601–610 (2014)
10. Google Knowledge Graph API: <https://developers.google.com/knowledge-graph/> (2018)
11. PAN 2014: <http://pan.webis.de/clef14/pan14-web/author-profiling.html> (2014)
12. Rangel, F., Chugur, P.R.I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: Overview of the 2nd author profiling task at pan 2014. In: Proceedings of the Conference and Labs of the Evaluation Forum (2014)
13. Salton, G., McGill, M.: Introduction to Modern Information Retrieval, McGraw Hill (1983)
14. Salton, G.: Automatic Text Processing: The Transform, Analysis, and Retrieval of Information by Computer. Addison-Wesley Longman Publishing Co. Inc. (1989)
15. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional: n-gramas sintácticos, filtrados y generalizados. 166 p. (2013)
16. Lan, M., Tan, C.L., Low, H.B., Sung, S.: A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In: Special interest tracks and posters of the 14th international conference on World Wide Web (WWW'05). (ACM), pp. 1032–1033 (2005)
17. Cilibrasi, R.L., Vitányi, M.B.P.: The Google Similarity Distance. In: IEEE Transactions on knowledge and data engineering, 19(3) (2007)
18. Sebastiani, F.: Text Categorization. In: Alessandro Zanasi (ed.), Text Mining and its Applications, WIT Press (2005)
19. Ferragina, P., Piccinno, F., Santoro, R.: On Analyzing Hashtags in Twitter. In: International (AAAI) Conference on Web and Social Media (2015)
20. Manning, C.D., et. al.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)



21. Van Rossum, G.: Python tutorial. In: Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI) (1995)
22. Pedregosa et al.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825–2830 (2011)
23. Villegas, M.P., Garcarena-Ucelay, M.J., Fernández, J.P., Álvarez-Carmona, M.A., Errecalde, M.L., Cagnina, L.: Vector-based word representations for sentiment analysis: a comparative study. In: XXII Congreso Argentino de Ciencias de la Computación, (CACIC), pp. 785–793 (2016)
24. Kocher, M., Jacques, S.: Distance measures in author profiling. In: *Information Processing & Management*, 53(5), pp. 1103–1119 (2017)
25. Álvarez-Carmona, M.A., López-Monroy, A.P., Montes-y-Gómez, M., Villaseñor-Pineda, L., Jair-Escalante, H.: INAOE's Participation at PAN'15: Author Profiling task. In: *Working Notes Papers of the CLEF* (2015)



# Identificación de relaciones taxonómicas de dominio usando métricas textuales

Yuridiana Alemán, María Somodevilla, Darnes Vilariño

Benemérita Universidad Autónoma de Puebla (BUAP),  
Facultad de Ciencias de la Computación,  
México

{yuridiana.aleman,mariajsomodevilla,dvilarinoayala}@gmail.com

**Resumen.** El proceso de aprendizaje de ontologías comprende tres pasos fundamentales: creación de clases y relaciones, población y evaluación. Este documento se enfoca en la creación de clases y relaciones, realizando un estudio sobre la detección de subclases para la ontología. Como caso de estudio se seleccionó un dominio pedagógico, donde se construyó un corpus semiautomático, a partir de artículos escritos en español publicados en el área de Ciencias Sociales. Para la detección de subclases fueron implementadas cuatro métricas de similitud textual basadas en términos, con las cuáles se construyó una heurística para determinar cuáles de los conceptos tienen posibilidades de convertirse en una subclase de la ontología y tienen una relación taxonómica con la clase principal. La evaluación se realizó mediante un *gold* verificado por un experto en el dominio y el contexto teórico de las clases y se obtuvo el recuerdo de cada clase. Los resultados muestran un recuerdo de 100 % 72 % y 67 % respectivamente para cada una de las clases, además de que se recuperaron conceptos relacionados a la clase principal mediante relaciones no taxonómicas.

**Palabras clave:** métricas semánticas, ontología, conceptos principales, relación *Is\_a*, pedagogía, dice, Jaccard, traslape, coseno.

## Domain Taxonomical Relationships Identification Using Textual Metrics

**Abstract.** The ontology learning process comprises three fundamental steps: creation of classes and their relationships, population and evaluation. This document focuses on the first step, by performing a study on the detection of ontology subclasses. As a case study was selected a pedagogical domain. A semiautomatic corpus, based on articles written in Spanish published in Social Sciences journals was built. Four textual similarity metrics based on terms for the detection of subclasses were implemented, with which a heuristic was conducted to determine which of these concepts have the potential to become in a subclass of the

ontology and at the same time keep a taxonomic relationship with the superclass. The evaluation was carried out through a gold, which was validated by an expert in the theoretical context domain. The results show a recall of 100 % ,72 % and 67 % for each class respectively. In addition, concepts related to the super classes were recovered through non-taxonomic relationships.

**Keywords:** semantic metrics, ontology, class, is\_a relation, pedagogy, dice, Jaccard, overlap, cosine.

## 1. Introducción

La información disponible en diversos repositorios va en aumento, especialmente en la Web. Por lo tanto, es necesario implementar técnicas para procesar esa información y relacionarla con otros repositorios a fin de incrementar el conocimiento extraído. Las ontologías se presentan como una opción para procesar esta información, que se pueden utilizar para gestión del vocabulario, aplicaciones de procesamiento del lenguaje natural, búsquedas, sistemas de recomendación, e-learning, entre otros [5]. El proceso de aprendizaje de la ontología integra la detección de clases, la creación, la población y la evaluación [6].

Este documento se centra en el primer paso del proceso de aprendizaje ontológico: la detección de clases. En investigaciones previas se trabajó con la detección y validación de clases principales, por lo que este artículo se centra en la detección de las subclases y relaciones entre conceptos. Para los experimentos se utiliza un corpus formado por documentos pedagógicos en español. El dominio pedagógico es extenso, por lo que la investigación consiste en la creación de herramientas que respalden las clases de los profesores en el aula. El corpus contiene tres clases principales: estilos de aprendizaje, tipos de inteligencias y estrategias de enseñanza-aprendizaje. Cada una de estas clases principales se subdivide de acuerdo a enfoques teóricos propuestos en la literatura pedagógica. El método propuesto utiliza métricas de similitud textual para extraer los términos más relacionados con cada una de las clases principales, tomando estos como subclases. En el proceso también se recuperan términos importantes que figuran como relaciones no taxonómicas con la clase principal.

El artículo está organizado en siete secciones que se describen a continuación. La sección 2 presenta los trabajos relacionados con la detección de clases y relaciones en el proceso de construcción de ontologías. La sección 3 describe teóricamente las ontologías, así como las tres clases principales del corpus. La sección 4 presenta las métricas de similitud textual utilizadas en los experimentos. La sección 6 presenta la metodología propuesta y la sección 7 muestra el análisis de los resultados. Finalmente, la sección 8 presenta las conclusiones y el trabajo futuro de la investigación.

## 2. Trabajos relacionados

Es importante mencionar que antes de iniciar el proceso de extracción de elementos principales, se debe tener un corpus para el dominio a trabajar, por lo que se analizaron algunas investigaciones sobre la construcción de corpus en diferentes dominios. EL trabajo que se discute en [12] se centra en la creación de un corpus lingüístico relevante escrito en lengua serbia, en dicha investigación, el enfoque es el análisis de sentimientos de los contenidos generados por los estudiantes en la educación superior. En el trabajo realizado en [22] se analizó el problema de crear un corpus de referencia para la clasificación de artículos de noticias en escenarios de etiquetas múltiples. Los autores proponen un enfoque semiautomático para crear un corpus de referencia que utiliza tres métodos de clasificación auxiliares: máquinas de vectores de soporte, clasificadores de vecinos más cercanos y otro basado en un diccionario.

En investigaciones como [20] se presentan métodos para la extracción de clases de manera semiautomática, utilizan una base de datos de verbos, alternancias de diátesis y esquemas sintáctico-semánticos del Español (ADESSE) [7] la cual contiene aproximadamente 160,000 cláusulas recuperadas de un corpus; con la ayuda de ADESSE se extraen patrones semánticos que llevan a la determinación de las clases para una ontología. Esta metodología fue aplicada en un subdominio educativo y replicada en ámbito financiero [19]. La extracción de clases se complementó con la opinión de expertos en el dominio. En [16] se presenta un método para la extracción de conceptos utilizando extracción de patrones lingüísticos y cálculo de pesos con métricas de procesamiento de lenguaje natural como el etiquetado morfológico.

Dentro del ámbito pedagógico, en la investigación de [11] se propone el proyecto OURAL (*Ontologies for the Use of digital learning Resources and semantic Annotations on Line*) el cual integra las disciplinas de ciencias de la educación, informática y psicología cognitiva con el fin de crear servicios para *E-learning*. Como resultados, se muestran las clases obtenidas mediante la aplicación de técnicas de PLN a situaciones de aprendizaje descritas en lenguaje natural. Este dominio se analiza también en [6], sin embargo, al ser aplicado al idioma Chino, utilizan un preprocesamiento para analizar las características de dicho idioma: acoplamiento, relevancia y consenso.

Otras investigaciones se centran en la educación en línea como [3], [4], [15] y recientemente [14], donde las ontologías se definen manualmente a partir de recursos XML disponibles en Internet, y la evaluación también es un proceso manual. Investigaciones como [23] se enfocan en el aprendizaje automático; en este trabajo, se crea una ontología basada en Internet de las cosas utilizado en un aula, teniendo en cuenta las inteligencias estudiantiles. [18] propone utilizar un modelo ontológico para la personalización del aprendizaje que involucre el perfil de los estudiantes de acuerdo con la teoría de inteligencia múltiple de Howard Gardner, así como usar una ontología de dominio que ayude a representar el conocimiento en plataformas de aprendizaje virtuales.

### 3. Ontologías de dominio

En las ciencias computacionales, una ontología se define como una especificación formal de una conceptualización [13]. Es una base de datos que describe los conceptos en el mundo o algún dominio, algunas de sus propiedades y cómo los conceptos se relacionan entre sí [24]. Esta base de datos se define a partir de un corpus base, en donde se extraen los elementos principales o palabras clave. Posteriormente, del mismo texto se infieren las relaciones entre palabras clave, de esta manera, se crea una estructura de grafo donde los nodos son las palabras clave y las aristas representan la relación existente entre ellas.

Entre las aplicaciones más representativas de las ontologías se encuentran la representación formal del conocimiento, lo que facilita el manejo e integración de datos con estructuras diferentes. Formalmente, una ontología se define como la sextupla  $O = (C, H, I, R, P, A)$  [5] donde:

- $C$  es el conjunto de entidades de la ontología,
- $H$  son las relaciones taxonómicas entre los conceptos,
- $I$  indica las relaciones entre instancias,
- $R$  es el conjunto de relaciones no taxonómicas,
- $P$  es el conjunto de propiedades de la ontología,
- $A$  representa el conjunto de axiomas y reglas que prueban la consistencia de la ontología que realizan el proceso de inferencia.

Una relación de ontología es una formalización de la manera en que las entidades están asociadas. La relación que se analiza en esta investigación es la de tipo *Is-a*, la cual es un vínculo entre clases en forma de jerarquía, una columna vertebral de una ontología. La organización jerárquica de las entidades, siempre por relaciones *Is-a*, permite la herencia de propiedades y la estructuración de la taxonomía.

#### 3.1. Dominio pedagógico

Dado que el dominio pedagógico es muy extenso, se establecieron tres clases principales a fin de obtener una herramienta de apoyo para el docente en clases presenciales.

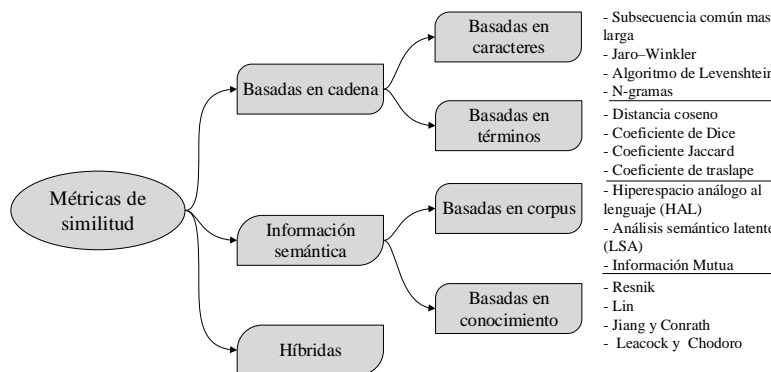
**Estilos de aprendizaje.** Los estilos de aprendizaje reflejan la forma en que el individuo aprende. Existen variaciones en cuanto a la manera en que los seres humanos captan y procesan información. Se han propuesto varias teorías para describir los distintos tipos de aprendizaje, para esta investigación se tomó como referencia el modelo de David Kolb [17], en el cual se determina un estilo de aprendizaje usando una escala denominada *Learning Style Inventory* (LSI). La teoría propone un método para describir cómo los estudiantes resuelven sus problemas y aplican conocimientos nuevos a partir de la experiencia personal dentro de su entorno de aprendizaje. Considera los procesos psicológicos de percepción y procesamiento [21]. El método propone 4 estilos de aprendizaje: activo, reflexivo, pragmático y teórico.

**Inteligencias múltiples.** La inteligencia se define como “la capacidad de resolver problemas o de crear productos que sean valiosos en uno o más ambientes culturales” [8]. Los seres humanos poseen una gama de capacidades y potenciales que se pueden emplear de muchas maneras productivas, tanto juntas como por separado, esta idea da origen a las inteligencias múltiples, las cuales han sido identificadas por Gardner: lógico-matemática, lingüística, espacial, musical, corporal, intrapersonal, interpersonal y naturalista.

**Estrategias de aprendizaje.** Una estrategia de aprendizaje es un conjunto de procedimientos que un alumno usa de manera consciente, controlada e intencional como herramientas flexibles para aprender y resolver problemas [1], también pueden ser definidas como conductas y pensamientos que un aprendiz utiliza durante el aprendizaje con la intención de influir en su proceso de codificación [25]. Aunque existen muchos enfoques para la clasificación de las estrategias de aprendizaje, [10] menciona tres principales tipos de estrategias: cognitivas, metacognitivas y las estrategias de manejo de recursos.

#### 4. Métricas de similitud textual

La tarea de similitud textual se encarga de comparar textos para conocer el parecido entre ellos. Para lograr este objetivo, se han propuesto en la literatura métricas que comparan la proximidad entre las palabras o caracteres de dos textos.



**Fig. 1.** Clasificación de las métricas de similitud textual según [3 y 10].

La Figura 1 muestra la clasificación propuesta por dos autores. Se presentan 3 clases principales: métricas basadas en cadenas, basadas en información semántica e híbridas.

Las métricas basadas en cadena contienen los enfoques basados en caracteres y en términos, mientras que las basadas en información semántica integran las métricas basadas en corpus y basadas en conocimiento.

Para la presente investigación se trabajan con las métricas basadas en términos. Las métricas basadas en caracteres pierden información al manejar un corpus lematizado; las métricas basadas en corpus suelen obtener resultados altos, pero son costosas en su implementación y necesitan un corpus extenso para calcular el valor de la co-ocurrencia de cada par de palabras [2]. Las métricas basadas en conocimientos están basadas en *WordNet*, y son utilizadas para el idioma inglés, por lo que no son pertinentes para esta investigación. Las métricas basadas en términos solo necesitan el corpus de entrada, aunque a mayor tamaño del texto se espera más exactitud en los resultados, aun así requieren menos recursos que las basadas en corpus. Las métricas más citadas en la literatura se describen en los siguientes párrafos.

**Coefficiente de Jaccard.** Se obtiene al dividir la intersección de términos entre la unión de los mismos. Su fórmula se presenta en la ecuación 1 y da por resultado el grado de superposición de dos conjuntos, en este caso, de dos N-gramas:

$$sim_J(t_1, t_2) = \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|}. \quad (1)$$

**Coefficiente de Dice.** Se basa en la teoría de conjuntos. Toma el número de las palabras que comparten ambas cadenas y los divide entre el número total de la suma de las palabras del texto uno y dos. Su cálculo está determinado por la ecuación 2. El coeficiente Dice da el doble de peso a las coincidencias positivas entre los términos. El resultado está normalizado entre cero y uno donde cero es nula similitud, mientras que uno se refiere a la máxima similitud [1]:

$$sim_D(t_1, t_2) = 2 \frac{|t_1 \cap t_2|}{|t_1| + |t_2|}. \quad (2)$$

**Coefficiente de Traslape.** Considera la cardinalidad de caracteres del texto más pequeño en lugar de la unión de los caracteres [9]. Para esta métrica, una coincidencia completa de dos cadenas es cuando una es un subconjunto de la otra, ecuación 3:

$$sim_T(t_1, t_2) = \frac{|t_1 \cap t_2|}{\min(|t_1|, |t_2|)}. \quad (3)$$

**Coefficiente de Coseno.** Se obtiene dividiendo la cardinalidad de la unión de los dos conjuntos entre la raíz cuadrada del producto de las cardinalidades de los conjuntos considerados, ecuación 4:

$$sim_C(t_1, t_2) = \frac{|t_1 \cap t_2|}{\sqrt{|t_1| |t_2|}}. \quad (4)$$

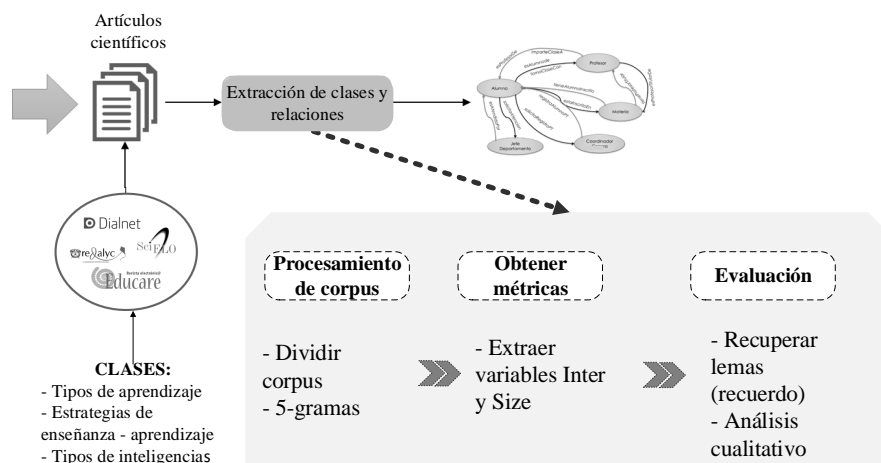
El coeficiente de Dice y de traslape son similares al coeficiente Jaccard, solo que el coeficiente de Dice da el doble de peso a las coincidencias positivas entre los



términos, y el coeficiente de traslape considera solo la cardinalidad de caracteres del texto más pequeño en lugar de la unión de los caracteres, como lo hace el coeficiente de Jaccard.

## 5. Metodología

La Figura 2 muestra la propuesta para la obtención de las subclases y las relaciones en el corpus. En la parte superior se muestra la metodología general, mientras que la flecha segmentada dirige al método seguido para esta investigación.

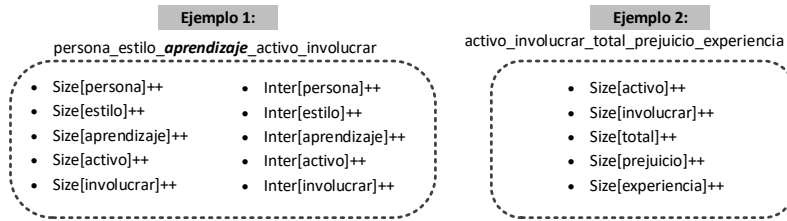


**Fig. 2.** Metodología general y proceso realizado para la obtención de subclases.

Previo a esta investigación, se realizó la validación de clases principales ( $C$ ) mediante técnicas de agrupamiento, donde  $C = \{TiposInteligencias, EstilosAprendizaje, EstrategiasEnseñanza\}$ . Para esto, se recolectó de manera manual un corpus compuesto por 51 artículos obtenidos de fuentes como Dialnet, Redalyc, SciELO y Educare.

El método propuesto tiene tres fases principales: En la primera, se tiene como entrada el corpus procesado (sin palabras cerradas ni signos de puntuación y lematizado). Dicho corpus se separa de acuerdo a la clase principal obteniendo 3 corpus diferentes de 17 instancias cada uno. Las métricas utilizadas en este experimento nos dicen en un rango de 0 a 1 que tan similar es una oración a otra, sin embargo, en este caso se utilizarán para determinar qué tan relacionado está un concepto respecto a otro, por lo que en lugar de utilizar oraciones completas se extraen 5-gramas y estos se toman como unidad de análisis.

En la segunda fase, se obtienen las variables necesarias para el cálculo de las métricas. Todas las métricas implementadas están basadas en el traslape de dos palabras a analizar:  $t_1$  y  $t_2$ , en este caso,  $t_1$  representa la clase principal y  $t_2$  cada uno de los lemas del vocabulario. Se extrajeron dos variables:  $Inter = x_1, x_2, x_3, \dots, x_n$  y  $Size = x_1, x_2, x_3, \dots, x_n$  donde  $Inter$  es el número de n-gramas en los que el lema y la clase principal coinciden,  $Size$  es el número de n-gramas en los que el lema aparece, independientemente si coincide o no con la clase,  $x$  representa cada uno de los lemas y  $n$  es el total del vocabulario para el corpus analizado. La Figura 3 muestra un ejemplo de cómo se obtienen estos vectores en dos n-gramas, uno en el que aparece la clase y otro en el que no. Como se puede observar, en el n-grama del ejemplo 1 aparece la clase (*Estilos de aprendizaje*) por lo que los dos vectores incrementan su valor para las palabras que aparecen en el n-grama, en el ejemplo 2 no aparece la clase, por lo que solo se incrementan los valores correspondientes al vector  $Size$ .



**Fig. 3.** Dos ejemplos en los que se obtienen los valores de los vectores  $Inter$  y  $Size$  para la clase *Estilos de aprendizaje*.

La tercera fase consiste en determinar cuáles lemas serán tomados como resultados, ya que los cálculos se hacen para todo el vocabulario. La hipótesis que se plantea es que si un lema tiene una similitud alta con la clase principal, significa que tiene una relación con esta, por lo que es una subclase para la ontología. Como se determinarán 4 métricas, se estableció un tope de 0.03 para determinar que un lema está relacionado con la clase, por lo que si un lema obtiene más de 0.03 en dos o más métricas, esta se toma como resultado, si no, se desecha. El tope es pequeño por el tamaño del corpus, aunque estas métricas suelen obtener resultados altos, el tamaño del corpus no es suficiente como para esperar resultados mayores al 50 %. El total de subclases recuperadas se evalúa utilizando el recuerdo, el cual es una métrica de recuperación de información que representa la fracción de los datos relevantes que son recuperados (Ecuación 5). Finalmente, se presenta un análisis cualitativo sobre los lemas relevantes recuperados por clase de acuerdo al tipo de relación que guardan con la clase principal:

$$R = \frac{\text{items relevantes recuperados}}{\text{items relevantes}}. \quad (5)$$

## 6. Resultados

Los resultados finales se muestran en la Tabla 1, en donde se observan el total de subclases a recuperar, las subclases recuperadas, el recuerdo por clase y otros conceptos que se recuperaron. Estos conceptos no son subclases, pero tienen relaciones no taxonómicas con la clase principal, ya sea que forman parte de la definición, aplicación o autor de dicho referente teórico.

**Tabla 1.** Análisis de los lemas obtenidos por cada clase.

Clase	Subclases		Otras	
	Reales	Recuperadas	Recuerdo	relaciones
Tipos de inteligencias	8	5	0.625	9
Estrategias de enseñanza	3	2	0.667	10
Estilos de aprendizaje	4	4	1.00	8

Sólo la clase de *Estilos de aprendizaje* tuvo un recuerdo del 100 % recuperando a las dos clases restantes, *Tipos de inteligencias* recuperó cinco de ocho, bajando su recuerdo a 63 %. En cuanto a la clase de *Estrategias de enseñanza*, solo se recuperaron dos de las tres subclases, por lo que obtuvo un recuerdo de 67 %. La clase *Estilos de aprendizaje* es la más estructurada ya que sólo cuenta con cuatro subclases, por lo que están muy bien definidas, en cuanto a *Tipos de inteligencias*, aunque la clasificación propuesta por Gardner es definida y cada una de las inteligencias propuestas tiene fundamentos y características propias, en cuanto a tareas de procesamiento de lenguaje natural, es complicado manejar siete clases distintas, esto hace que en los experimentos no se separen completamente las instancias y no se recuperen la totalidad de las clases. Además, puede que alguno de los artículos se enfoque en solo una o dos inteligencias, por lo que el vocabulario asociada a estas es mayor al vocabulario de otras menos mencionadas en los artículos. La clase *Estrategias de enseñanza* presenta una clasificación muy general y en muchas ocasiones los autores usan nombres específicos para nombrarlas, no solo como metacognitivas, cognitivas o de apoyo.

En las siguientes tablas se presentan los resultados por clase. Se anexa el valor del vector *Inter*, ya que este sirve para determinar en cuantos n-gramas coincide cada lema y la clase, y los valores de cada una de las métricas, donde los resultados van de 0 a 1. Solo se muestran los lemas que cumplen la heurística planteada: Obtener más de 0.03 en al menos dos métricas de las calculadas. Los lemas que son considerados como subclases se muestran en negritas, mientras que los que tienen otro tipo de relación con la clase principal se muestran en itálicas.

En la Tabla 2 se muestran los resultados para la clase *Tipos de inteligencias*. Como se menciona en la sección 3.1, este enfoque teórico distingue ocho tipos de inteligencias, de las cuales cinco aparecen en negritas en la tabla: musical, lingüística, interpersonal, emocional y lógico-matemática. Las inteligencias de tipo espacial, intrapersonal y naturalista no aparecen en la lista. Como se mencionó antes, si existen muchas subclases es complicado determinar las características

específicas de cada una de ellas. Además, estos resultados dependen totalmente del corpus y es posible que en algunos artículos se mencionen los 8 tipos de inteligencias, pero los experimentos o las discusiones se centren en las más predominantes dentro de la muestra estudiada. En la mayoría de los casos, los coeficientes de traslape y coseno tienen valores mayores a los coeficientes de Dice.

**Tabla 2.** Resultados para N-gramas en la clase *Tipos de inteligencias*.

Lema	-Intersección-	Dice -	Jaccard -	Traslape -	Coseno -
inteligencia	5702	1.0000	1.0000	1.0000	1.0000
múltiple	1332	0.3589	0.2187	0.774	0.4252
poder	163	0.0456	0.0234	0.1131	0.0569
numero	211	0.0611	0.0315	0.1758	0.0807
teoría	418	0.1231	0.0656	0.3835	0.1677
desarrollar	214	0.0632	0.0326	0.1998	0.0866
capacidad	170	0.0507	0.0260	0.1685	0.0709
ser	105	0.0314	0.0159	0.1062	0.0442
gardner	233	0.0698	0.0361	0.2385	0.0987
alumno	139	0.0421	0.0215	0.1544	0.0614
cada	209	0.0643	0.0332	0.2606	0.0977
diferente	153	0.0474	0.0243	0.2032	0.0738
persona	105	0.0328	0.0167	0.1491	0.0524
relación	235	0.0735	0.0381	0.3381	0.1180
considerar	104	0.0331	0.0168	0.1781	0.0570
rendimiento	159	0.0510	0.0262	0.2983	0.0912
tipo	294	0.0944	0.0495	0.5558	0.1693
musical	103	0.0332	0.0169	0.2060	0.0610
lingüístico	231	0.0747	0.0388	0.4793	0.1393
interpersonal	105	0.0340	0.0173	0.2253	0.0644
emocional	168	0.0552	0.0284	0.4386	0.1137
basar	120	0.0394	0.0201	0.3141	0.0813
todo	97	0.0321	0.0163	0.2820	0.0693
lógicomatemática	156	0.0521	0.0267	0.5379	0.1213

En cuanto a los otros lemas relacionados con la clase principal, se recuperaron 9 lemas que describen o forman parte del concepto de *Tipos de inteligencias*: La *teoría* de la inteligencias *múltiples* fue propuesta por *Gardner*, cada *persona* o *estudiante* tiene un *tipo* de inteligencia dominante, con la cual *desarrolla capacidades* diferentes, el correcto manejo de estas inteligencias permite aumentar el *rendimiento* académico. Si bien estos términos no son relaciones taxonómicas, son importantes para describir a la clase principal. Es importante mencionar que en esta clase es que las palabras que describen a la clase obtuvieron tienen más intersecciones que las subclases, pero en el resultado de las métricas, a excepción del coeficiente Jaccard, los resultados de las subclases son más altos que los de estos lemas.

La Tabla 3 muestra los resultados para la clase *Estilos de aprendizaje*. De acuerdo a la sección 3.1 existen cuatro tipos de aprendizaje, y todos estos se recuperan en la lista de la tabla: reflexivo, activo, teórico y pragmático. Al igual que en la clase anterior, con el coeficiente de Traslape se obtuvieron los resultados más altos. En cuanto a los otros lemas recuperados, algunos son parte de las características de esta clase por ejemplo: Para determinar el *estilo* de aprendizaje en un *estudiante*, se utiliza un *cuestionario* propuesto por *Honey-Alonso* llamado *CHAEA* (Cuestionario de Honey y Alonso de Estilos de

Aprendizaje). El conocer el estilo de aprendizaje también puede ir encaminado a la mejora del *rendimiento académico* del *alumno*, para implementar *estrategias* que ayuden al aprendizaje significativo. En esta clase, no se muestra una división entre las relaciones taxonómicas y no taxonómicas, las subclases se encuentran distribuidas a lo largo de la tabla.

**Tabla 3.** Resultados para N-gramas en la clase *Estilos de aprendizaje*.

Lema	-Intersección-	Dice -	Jaccard -	Traslape -	Coseno -
aprendizaje	6160	1.0000	1.0000	1.0000	1.0000
estilo	3903	0.5661	0.3948	0.6336	0.5694
estudiante	717	0.1616	0.0879	0.264	0.1753
estrategia	509	0.1422	0.0765	0.5095	0.2052
chaea	394	0.1076	0.0568	0.3382	0.1471
académico	312	0.0827	0.0431	0.2254	0.1069
reflexivo	266	0.0725	0.0376	0.2268	0.0990
activo	254	0.0688	0.0356	0.2077	0.0925
cuestionario	227	0.0641	0.0331	0.2459	0.0952
rendimiento	214	0.0607	0.0313	0.2415	0.0916
alumno	212	0.0580	0.0299	0.1839	0.0795
Vol	196	0.0524	0.0269	0.1492	0.0689
número	191	0.0501	0.0257	0.1306	0.0636
teórico	190	0.0527	0.0270	0.1798	0.0745
preferencia	184	0.0522	0.0268	0.2081	0.0788
proceso	175	0.0509	0.0261	0.2441	0.0833
alonso	158	0.0467	0.0239	0.2607	0.0818
universitario	154	0.0446	0.0228	0.2064	0.0718
análisis	140	0.0403	0.0206	0.1768	0.0634
promedio	134	0.0382	0.0195	0.1582	0.0587
pragmático	126	0.0361	0.0184	0.1518	0.0557
octubre	116	0.0361	0.0184	0.4265	0.0896
estudio	113	0.0317	0.0161	0.1177	0.0465
honey	111	0.0333	0.0169	0.2216	0.0632
relación	105	0.0319	0.0162	0.2530	0.0657
variable	104	0.0308	0.0157	0.1772	0.0547
predominante	102	0.0322	0.0163	0.5543	0.0958

Finalmente, la Tabla 6 muestra los resultados para la clase *Estrategias de enseñanza*, donde se pueden observar que se recuperaron dos de las tres subclases: cognitiva y metacognitiva. Aunque esta clase cuenta con una clasificación, esta no es adoptada por todos los autores, e incluso, aparte de esta clasificación, cada estrategia tiene un nombre, independientemente de la subclase a la que pertenezca. Por ejemplo, una estrategia para comprensión de lectura puede ser una estrategia de manejo de recursos, pero los autores se refieren a ella como estrategia de lectura, de hecho, la palabra lectura está en la lista de los lemas recuperados, pero al no ser parte de la clasificación, no se toma como relación taxonómica. Esta riqueza de nombres hace que se dificulte localizar estas subclases en el corpus.

Para esta clase también se recuperaron algunos conceptos que tienen relaciones no taxonómicas con el concepto principal como *aprendizaje*, *enseñanza*, *conocimiento*, *planificación*, *proceso*, *motivación*. Estos conceptos ayudan a la descripción de una estrategia de enseñanza aprendizaje como una herramienta para que el estudiante adquiera conocimiento, estas estrategias son planificadas

**Tabla 4.** Resultados para N-gramas en la clase *Estrategias de enseñanza aprendizaje*.

Lema	-Intersección-	Dice -	Jaccard -	Traslape -	Coseno -
estrategia	3803	1.0000	1.0000	1.0000	1.0000
aprendizaje	867	0.2289	0.1292	0.2297	0.2289
<b>cognitivo</b>	524	0.1957	0.1085	0.3378	0.2158
<b>metacognitivas</b>	410	0.1827	0.1005	0.5985	0.2540
estudiante	214	0.0656	0.0339	0.0785	0.0665
uso	203	0.0886	0.0463	0.2599	0.1178
conocimiento	176	0.0666	0.0345	0.1188	0.0742
emplear	175	0.0803	0.0418	0.3142	0.1202
utilizar	170	0.0760	0.0395	0.2534	0.1064
categoría	146	0.0668	0.0346	0.2575	0.0994
poder	128	0.0495	0.0254	0.0936	0.0561
estilo	116	0.0571	0.0294	0.4514	0.1173
autorregulación	106	0.0501	0.0257	0.2482	0.0832
motivación	106	0.0491	0.0251	0.2046	0.0755
enseñanza	104	0.0457	0.0234	0.1387	0.0616
permitir	89	0.0384	0.0196	0.1075	0.0502
proceso	88	0.0317	0.0161	0.0502	0.0341
numero	83	0.0315	0.0160	0.0568	0.0352
patrón	80	0.0397	0.0202	0.3493	0.0857
frecuencia	80	0.0389	0.0198	0.2564	0.0734
enfocar	76	0.0386	0.0197	0.5547	0.1053
elaboración	74	0.0369	0.0188	0.3610	0.0838
nivel	74	0.0323	0.0164	0.0956	0.0431
utilización	71	0.0354	0.0180	0.3333	0.0789
significativo	70	0.0319	0.0162	0.1207	0.0471
deber	69	0.0306	0.0155	0.0980	0.0422
tipo	68	0.0312	0.0158	0.1216	0.0466
usar	65	0.0322	0.0164	0.2778	0.0689
lectura	64	0.0303	0.0154	0.1509	0.0504
planificación	63	0.0307	0.0156	0.2059	0.0584

por los docentes procurando incentivar la motivación de los estudiantes, por ejemplo, las estrategias para la comprensión de lectura.

## 7. Conclusiones y trabajo futuro

En este artículo se presentó el análisis de métricas de similitud basadas en término, a fin de determinar las subclases para la construcción de una ontología. Para los procedimientos se utilizó un corpus pedagógico con tres clases principales. En los resultados se muestran los lemas recuperados por clase, entre los que se encuentran subclases, conceptos descriptivos y algunas palabras no relacionadas con la clase o que aportan poca información para la ontología.

La clase de Estilos de aprendizaje recupera todas las subclases, y los otros conceptos recuperados describen bien a la clase. En la clase Estrategias de aprendizaje solo se recuperan dos de tres mientras que en la clase Tipos de inteligencia se recuperan cinco de ocho; estas clases no presentan subclases bien definidas, por lo que el tamaño del corpus no permite establecer todas las subclases.

En cuanto a las métricas utilizadas, el coeficiente de traslape es el que obtuvo resultados más altos en las tres categorías, pero esta métrica puede ser engañosa, ya que si un lema tiene pocas apariciones, pero estas están en el mismo n-grama que la clase, el coeficiente de traslape es igual a 1. Es por eso que se utilizan las

otras métricas para recuperar los lemas. Los coeficientes de Dice y de Jaccard obtienen resultados muy bajos, incluso más que el tope establecido de 0.03. Esto tiene que ver con la longitud del corpus, que para efectos de estas métricas, puede considerarse pequeño. Sin embargo, a pesar de los resultados bajos en las métricas, los valores más altos se encuentran en los lemas importantes para la clase, ya sea como una subclase o como una relación no taxonómica.

Como trabajo futuro, se pretende incrementar el corpus a fin de obtener una mayor riqueza en el vocabulario. Al incrementar las instancias del corpus de entrada, los resultados en las métricas permitirán recuperar solo términos importantes. Además, se implementarán las métricas basadas en corpus.

## Referencias

1. Barriga, F., Hernández, G.: Estrategias docentes para un aprendizaje significativo. Una interpretación constructivista. McGraw Hill, México (2004)
2. Álvarez Carmona, M.n.: Detección de similitud semántica en textos cortos. Ph.D. thesis, Instituto Nacional de Astrofísica Óptica y Electrónica (2014)
3. Dai, X., Li, X.: Study of learning source ontology modeling in remote education. In: 2010 International Conference on Multimedia Technology. pp. 1–4 (Oct 2010)
4. Du, L., Zheng, G., You, B., Bai, L., Zhang, X.: Research of online education ontology model. In: 2012 Fourth International Conference on Computational and Information Sciences. pp. 780–783 (Aug 2012)
5. Faria, C., Girardi, R.: A domain-independent process for automatic ontology population from text. Science of Computer Programming 95, Part 1, 26 – 43 (2014), <http://www.sciencedirect.com/science/article/pii/S0167642313003419>, special Issue on Systems Development by Means of Semantic Technologies
6. Fu, J., Jia, K., Xu, J.: Domain ontology learning for question answering system in network education. In: 2008 The 9th International Conference for Young Computer Scientists. pp. 2647–2652 (Nov 2008)
7. García-Miguel, J.M., Vaamonde, G., Domínguez, F.G.: Adesse, a Database with Syntactic and Semantic Annotation of a Corpus of Spanish. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (may 2010)
8. Gardner, H.: Estructuras de la Mente (Sep 2001)
9. Gomaa, W., Fahmy, A.: A survey of text similarity approaches 68 (04 2013)
10. González, M., Tourón, J.: Autoconcepto y rendimiento escolar: sus implicaciones en la motivación y en la autorregulación del aprendizaje. Eunsa (1992)
11. Grandbastien, M., Azouaou, F., Desmoulins, C., Faerber, R., Leclet, D., Quenu-Joiron, C.: Sharing an ontology in education: Lessons learn from the OURL project. In: Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007). pp. 694–698 (July 2007)
12. Grljevic, O., Bosnjak, Z.: Development of serbian higher education corpus. In: 2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI). pp. 177–181 (Nov 2015)
13. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. 43, 907–928 (December 1995), <http://portal.acm.org/citation.cfm?id=219666.219701>

14. Hssina, B., Bouikhalene, B., Merbouha, A.: An ontology to assess the performances of learners in an e-learning platform based on semantic web technology: Moodle case study. In: Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 103–112. Springer (2017)
15. Hu, J., Li, Z., Xu, B.: An approach of ontology based knowledge base construction for chinese k12 education. In: 2016 First International Conference on Multimedia and Image Processing (ICMIP). pp. 83–88 (June 2016)
16. Kang, Y.B., Haghighi, P.D., Burstein, F.: Cfindex: An intelligent key concept finder from text for ontology development. *Expert Systems with Applications* 41(9), 4494 – 4504 (2014), <http://www.sciencedirect.com/science/article/pii/S0957417414000189>
17. Kolb, D.: Learning style inventory. MA: Hay Group, Hay Resources Direct, Boston USA (1976)
18. Méndez, N.D.D., Carranza, D.A.O., Ocampo, M.G.: Representación ontológica de perfiles de estudiantes para la personalización del aprendizaje. *Revista Educación en Ingeniería* 10(19), 105–115 (2015)
19. Ochoa, J.L., Hernández-Alcaraz, M.L., Almela, A., Valencia-García, R.: Learning semantic relations from Spanish natural language documents in the financial domain. In: Proceedings of the 3rd International Conference on Computer Modeling and Simulation, held at Mumbai, India. Chengdu: Institute of Electrical and Electronics Engineers, Inc. pp. 104–108 (2011)
20. Ochoa, J.L., Hernández-Alcaraz, M.L., Valencia-García, R., Martínez-Bejar, R.: A semantic role-based methodology for knowledge acquisition from Spanish documents. *International Journal of Physical Sciences* 6(7), 1755–1765 (2011)
21. Olivos, P., Santos, A., Martín, S., Cañas, M., Gómez, E., Maya, Y.: The relationship between learning styles and motivation to transfer of learning in a vocational training programme. *Suma Psicológica* 23(1), 25–32 (2016)
22. Teixeira, J., Sarmiento, L., Oliveira, E.: Semi-automatic creation of a reference news corpus for fine-grained multi-label scenarios. In: 6th Iberian Conference on Information Systems and Technologies (CISTI 2011). pp. 1–7 (June 2011)
23. Uskov, V., Pandey, A., Bakken, J.P., Margapuri, V.S.: Smart engineering education: The ontology of internet-of-things applications. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 476–481 (April 2016)
24. Weigand, H.: A multilingual ontology-based lexicon for news filtering-the TREVI project. In: Proceedings of the IJCAI Workshop on Multilingual Ontologies-Nagoya (1997)
25. Weinstein, C.E., Mayer, R.E.: The teaching of learning strategies. In: Innovation abstracts. vol. 5. ERIC (1986)



# Sistema para la generación personalizada de resúmenes a partir de múltiples documentos

Orlando Hernández Hernández, Esaú Villatoro Tello,  
Christian Lemaître León

Universidad Autónoma Metropolitana (UAM) Unidad Cuajimalpa,  
Departamento de Tecnologías de la Información,  
México

orlandoox5@gmail.com,  
{evillatoro,clemaître}@correo.cua.uam.mx

**Resumen.** En el mundo actual las necesidades de información son cada vez mayores y al mismo tiempo muy diversas, esto último debido a los distintos perfiles de usuarios en la web. Durante la última década el aumento en la generación de contenidos ha crecido tanto que se requiere una gran cantidad de recursos para almacenar y procesar toda esta información. Esta explosión de información obliga a desarrollar herramientas inteligentes que faciliten la búsqueda, organización y recuperación de la información para los distintos usuarios. Dentro de este trabajo se describe el desarrollo de una herramienta para la generación automática de resúmenes de múltiples documentos, los cuales son guiados por la consulta de un usuario. La herramienta desarrollada emplea técnicas de Inteligencia Artificial para identificar los distintos sub-tópicos contenidos en una colección de documentos, con los cuales es posible construir un resumen que satisface las necesidades de información de usuarios específicos.

**Palabras clave:** resumen automático de múltiples documentos, resumen guiado por consulta, aprendizaje automático, agrupamiento de documentos, similitud de textos.

## User's Profile-Aware Multi-Document Summarization System

**Abstract.** Nowadays the information needs are constantly increasing and at the same time they are very diverse, this last due to the presence of users with different profiles. During the last decade the increase in content generation has grown so much that it requires a great amount of resources for storing and accurately processing all this information. This explosion of information requires the development of intelligent

tools that facilitates searching, organizing and retrieving information for different users. In this paper we describe the development of a tool for the automatic generation of summaries of multiple documents, which are guided by the query given from the user. Developed tool uses Artificial Intelligence techniques to identify the different subtopics contained in a collection of documents, then, the most relevant pieces of information are used for building a summary that satisfies the information needs of specific users.

**Keywords:** multiple document summarization, query focused summarization, machine learning, clustering, textual similarity.

## 1. Introducción

Actualmente la generación de contenido textual ha crecido de una manera exponencial, de tal forma que se hace una tarea casi imposible leer toda la información referente a uno o varios temas. La información que actualmente podemos encontrar en la red gracias a los buscadores, nos facilitan en gran medida la tarea de recuperación de información ya que el usuario solo debe de ingresar una consulta. Sin embargo, estos buscadores devuelven una lista muy grande de documentos relacionados a la consulta [1]. Debido a que los resultados de la búsqueda son demasiados, es tarea del usuario realizar la actividad de discriminar entre aquellos documentos que le son relevantes y los que no, pues es altamente probable que muchos de los documentos recuperados presentan información redundante, información desactualizada, o incluso algunos que no contengan información relevante para el usuario.

Herramientas como lo son los sistemas de generación de resúmenes pueden impactar de manera positiva en la problemática planteada. Un resumen es la síntesis concisa y coherente de la información más importante contenida en uno o más documentos, por lo que un sistema generador de resúmenes tiene como objetivo presentar al usuario las ideas principales de los documentos de referencia en un texto pequeño [2, 3]. Tradicionalmente, los sistemas de generación de resúmenes se caracterizan por proponer métodos eficientes para la detección de la información relevante de uno o varios documentos, así como identificar las redundancias de forma que es posible obtener un texto simplificado en el que se plantean las ideas clave del conjunto de documentos. Sin embargo, los sistemas de generación de resúmenes han tenido enfoques muy genéricos, es decir, suelen construir resúmenes similares para distintos usuarios. Ante la diversidad de usuarios y de necesidades de información, es que surgen los sistemas de generación de resúmenes guiados por consulta. El objetivo principal de este tipo de sistemas es construir un resumen que satisfaga las necesidades de información de usuarios específicos [2, 4, 5].

En este sentido, el presente trabajo describe el desarrollo de una herramienta para la generación de resúmenes de múltiples documentos guiados por consulta.

Nuestro software emplea técnicas de Inteligencia Artificial y de Procesamiento de Lenguaje Natural para la identificación de información redundante así como de la diversidad de los tópicos contenidos en una colección de documentos. Una característica relevante del sistema desarrollado es la incorporación de un módulo de recuperación de información, el cual es capaz de descargar información de la web, que posteriormente es procesada para la construcción del resumen. Esta funcionalidad permite al usuario obtener en tiempo real todos aquellos documentos que son relevantes a una necesidad de información, es decir, documentos que están relacionados temáticamente. Una vez descargados los documentos, el usuario podrá realizar uno o varios resúmenes que responden a necesidades específicas de información. Es importante mencionar la herramienta desarrollada tiene un modo de funcionamiento 'fuera-de-línea', con el cual el usuario puede proporcionar a la herramienta directamente los documentos que quiere analizar. Como se mostrará más adelante, el software desarrollado está listo para descargarse y ejecutarse en plataformas Windows y Linux.

El resto del documento está organizado de la siguiente manera. La sección 2 describe brevemente algunos de los trabajos relacionados, la sección 3 describe el método de generación de resúmenes de nuestro software; la sección 4 muestra algunas pantallas del sistema así como un ejemplo de cómo el sistema toma en cuenta las preferencias del usuario para su funcionamiento. Finalmente la sección 5 enuncia las conclusiones alcanzadas así como algunas líneas de trabajo futuro para la optimización del sistema desarrollado.

## **2. Trabajo relacionado**

La generación de resúmenes es una tarea que el PLN aborda por medio de distintos enfoques y métodos con la finalidad de mejorar la calidad de los resúmenes generados para satisfacer las necesidades de información del usuario. A continuación se describirán algunos trabajos precursores al nuestro.

En el trabajo descrito en [7], los autores emplean a los términos más frecuentes como los más relevantes en el texto, y por lo tanto son los que ayudan a determinar la temática principal de un texto. En [7], el autor solo extrae para la generación del resumen aquellas oraciones que contienen presencia de estos términos. Por otro lado, trabajos que han empleado técnicas de aprendizaje automático para la construcción de resúmenes son como el descrito en [5]. En este enfoque se utilizan atributos que describen a las oraciones en términos de su ubicación en el documento, número de palabras, etc. Con estos atributos los autores entrenan un modelo de aprendizaje computacional, con el cual logran generar resúmenes de un sólo documento. Estos dos trabajos, representan referentes clásicos de la generación de resúmenes de un sólo documento. Y muestran que, hasta cierto punto, atributos relacionados a la posición de las palabras, su frecuencia, su complejidad, similitud con el título, longitud de las oraciones, etc., son atributos que ayudan a identificar porciones de texto importantes para la construcción del resumen. Sin embargo, su funcionalidad no incorpora al usuario

en el proceso de la construcción, es decir, los resúmenes que se generan para un mismo documento serán similares sin importar el perfil del usuario.

Más recientemente, con el afán de incorporar las necesidades del usuario en la generación de los resúmenes, surgen propuestas que incorporan el perfil del usuario. Por ejemplo en [9], los autores proponen un método de regresión para la clasificación de las oraciones que podrían ser parte (o no) del resumen tomando como referencia una consulta dada por el usuario. Se utilizan siete características en total para la generación de resúmenes de múltiples documentos, tres son dependientes de la consulta: coincidencia nombre y entidad, la similitud de palabras y la coincidencia semántica. Las otras 4 son independientes de la consulta: la posición de la oración, entidad nombrada, el resultado de TF-IDF en las palabras y la penalización de palabras vacías<sup>1</sup>. Su sistema propuesto se necesita entrenar con una serie de resúmenes generados por humanos. Este método utiliza varios técnicas basadas en n-gramas<sup>2</sup> las cuales se encargan de calcular el puntaje de las oraciones para identificar las que son relevantes y las que no. Finalmente para eliminar la redundancia de información se utilizan MMR (Maximal Marginal Relevance), la cual es una medida para cuantificar desemejanza entre la oración que se está considerando y las que ya están seleccionadas.

De forma similar, en [11] se utiliza un modelo de aprendizaje supervisado, utilizando el corpus del DUC 2005 para entrenar al sistema. En general, el sistema propuesto consta de tres pasos para lograr la generación de resúmenes. El primero es la clasificación de oraciones en orden de relevancia de acuerdo a la consulta dada, en este paso se utilizaron dos algoritmos de clasificación: *Support Vector Regression* (SVR) [10] y *LambdaMART* [3]. En este paso se eliminan palabras vacías así como un conjunto de palabras específicas, las cuales son: “*discuss, describe, specify, explain, identify, include, involve, note*”. El segundo es un método de compresión de oraciones, y finalmente el tercer paso es el Post-procesamiento para la generación del resumen, para lo cual utilizan esquemas de pesado *TF-IDF*<sup>3</sup> para la identificación de los elementos relevantes. La desventaja principal de estos trabajos es que requieren de colecciones de datos etiquetados para poder aplicar técnicas de aprendizaje supervisado.

Por otro lado, en [4] se propone un método que no requiere de un conjunto de datos etiquetados. Los autores proponen un método que se compone de cuatro etapas. La primera es un proceso que busca identificar la similitud de los elementos de los documentos, para lo cual crea una matriz de similitudes, misma que involucra a la consulta. El siguiente paso es la ponderación, para esto se suma todas las filas correspondientes a cada documento, de esta forma se logra identificar (rankear) a los documentos más representativos. El tercer paso es ordenar los documentos en forma descendente y eliminar los que tengan

<sup>1</sup> Del ingles *stop words* son términos que no tienen carga semántica como los conectores, artículos, etc.

<sup>2</sup> Es una subsecuencia de n elementos de una secuencia dada

<sup>3</sup> Del inglés *Term Frequency-Inverse Document Frequency*, este esquema le da mayor relevancia a los términos que son menos frecuentes en la colección, pero más frecuentes en el documento

menor importancia. Finalmente se generará el resumen con los documentos más similares a la consulta.

Como se puede observar, la variedad de heurísticas utilizadas para la construcción de métodos de generación de resúmenes es muy variada. Sin embargo, algo que tienen en común estos trabajos es el uso de técnicas de PLN para identificar aquellas porciones de información que se asemejan a la necesidad de información del usuario. Inspirados por estos trabajos, nuestro sistema desarrollado emplea técnicas no supervisadas de aprendizaje, aspectos que le proporcionan una flexibilidad importante a nuestro sistema al no depender de un conjunto cerrado de etiquetas. Finalmente, es importante mencionar que para el desarrollo de nuestra herramienta de generación de resúmenes, tomamos como base las ideas propuestas en [8].

### 3. Método propuesto

La figura 1 muestra de manera esquemática los componentes principales del sistema de generación de resúmenes. En términos generales el sistema esta conformado por dos módulos: el sistema de Recuperación de Información, y el módulo de Generación del Resumen.

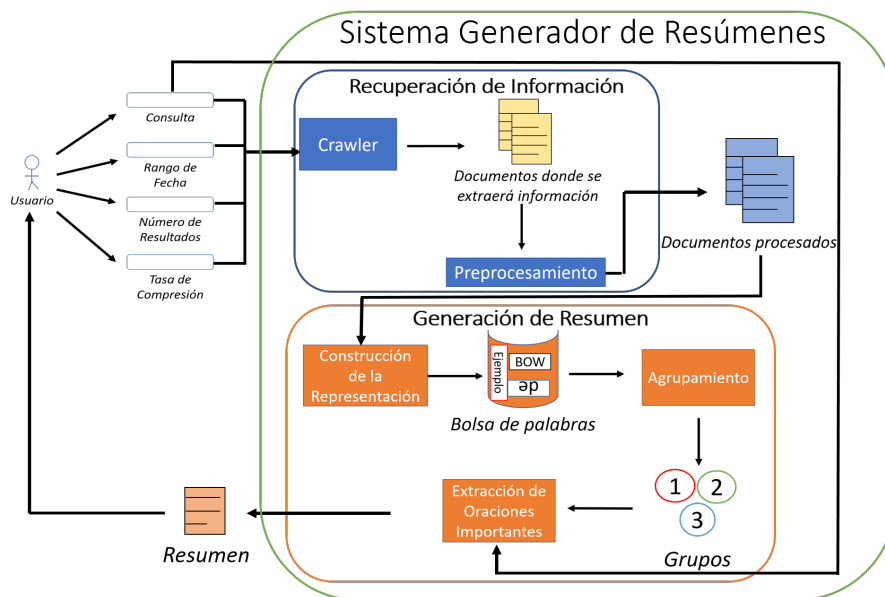


Fig. 1. Arquitectura general del sistema propuesto.

Como es posible observar en la figura 1, el sistema recibe como parámetros de entrada la 'consulta', 'rango de fechas', y la 'tasa de compresión' que deberá

contener el resumen. La consulta proporcionada sirve para que el módulo de recuperación de información descargue de la web todos aquellos documentos que se presumen relevantes a la consulta. El parámetro de rango de fechas sirve para delimitar la búsqueda, de igual forma el parámetro de ‘número de resultados’. Una vez concluida la descarga, el módulo de generación de resumen entra en acción. En su interior, éste utiliza técnicas de agrupamiento y de PLN para la construcción del resumen final. Note que para la construcción del resumen se utiliza la consulta dada por el usuario, sin embargo es importante mencionar que esta consulta puede variar con respecto a la consulta que provoco la descarga. A continuación se describirá con mayor detalle el funcionamiento de cada uno de los componentes del sistema desarrollado.

### 3.1. Recuperación de información

Nuestro módulo de Recuperación de Información (RI) tiene como objetivo la obtención de la información relevante a una consulta. Para la versión actual de nuestro sistema, la fuente de información donde se hace la búsqueda de información es el dominio del periódico ‘El Universal’<sup>4</sup>.

De esta forma, como cualquier sistema de RI, el principal parámetro de entrada es la consulta del usuario, sin embargo, con el afán de delimitar el proceso de búsqueda, dos parámetros adicionales son incorporados: rango de fechas, y número de resultados. El primero ayuda a establecer los rangos de fechas entre los cuales debieron haber sido publicados los documentos para ser recuperados. El segundo parámetro establece a través de un número específico la cantidad de documentos que se deben de recuperar. Si el usuario decide emplear el segundo parámetro, la descarga siempre se hará del documento más reciente al más antiguo hasta cumplir con el valor establecido.

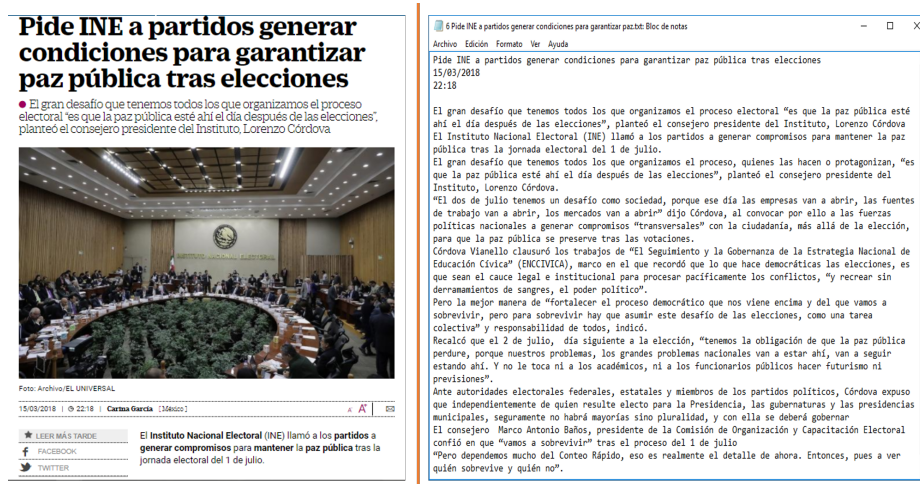
Específicamente, los submódulos que se desarrollaron para el motor de RI son: un crawler y un módulo de pre-procesamiento de información. A continuación describimos los objetivos de cada uno.

**Crawler.** Para el desarrollo del crawler fue necesario familiarizarse con la estructura del sitio de ‘El Universal’. Una vez realizado esto, lo que nuestro crawler hace es aprovechar el motor de búsqueda propio de la página del universal para localizar aquellas noticias que satisfacen la necesidad de información del usuario, es decir, se localizan todos aquellos documentos que contienen los términos de la consulta proporcionada por el usuario.

Una vez localizadas los documentos relevantes, el crawler navega por cada uno de estos documentos descargando solo aquellos que satisfacen la restricción de las fechas proporcionadas por el usuario, o si se da un número máximo de descargas, se descargan del más actual al más antiguo hasta cubrir el requisito. Es importante mencionar que el crawler descarga en el disco local solo el texto de cada documento, de momento, no se recuperan fotos y/o vídeos. Un ejemplo de una nota descargada por nuestro sistema es

<sup>4</sup> <http://www.eluniversal.com.mx/>

mostrado en la figura 2. Observe que del lado izquierdo se muestra la misma nota en su formato original en el sitio de El Universal.



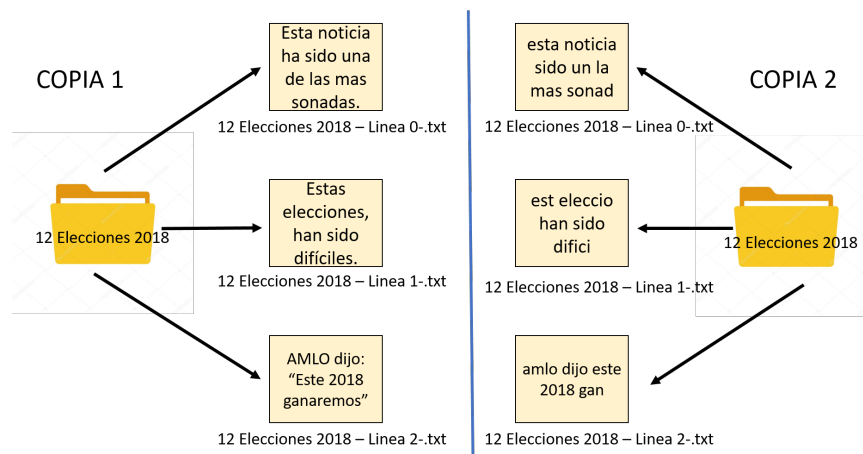
**Fig. 2.** Visualización de la noticia original(lado izquierdo) y la noticia descargada por el *Crawler*(lado derecho).

**Pre-procesamiento de la información.** Una vez que se han obtenido los documentos relevantes, este módulo hace un limpiado de la información quitando etiquetas XML, HTML o código Java Script. Cuando el documento ha sido limpiado el texto es guardado en documentos .txt, y son nombrados con el numero de descarga y el nombre de la noticia, ejemplo: *"12 Elecciones 2018.txt"* (sin comillas), indicando que es la noticia número 12 descargada. Internamente, el módulo de pre-procesamiento segmenta las noticias en oraciones<sup>5</sup> y genera dos copias de cada noticia (figura 3) segmentadas por oraciones. La copia 1 es utilizada para obtener las oraciones en su formato original, mientras que las oraciones almacenadas en la copia 2 pasan por un proceso de normalización, es decir truncado, llevadas a minúsculas, eliminación de símbolos de puntuación, etc. Los documentos en su versión de la copia 2 son la entrada del módulo de generación del resumen.

### 3.2. Generación de resumen

Este módulo es el encargado de procesar la información con la finalidad de construir un resumen que satisfaga las necesidades de información de un usuario en específico. Como se mostró en la figura 1, este módulo comprende

<sup>5</sup> Para hacer la segmentación se utilizó la función `sent_tokenize` proporcionada por la biblioteca de Python NLTK (<https://www.nltk.org>)



**Fig. 3.** Almacenamiento interno de la información en dos modalidades, información original (izq.) , e información pre-procesada para su posterior análisis (der.).

varios procesos: construcción de la representación, agrupamiento, y extracción de oraciones importantes. A continuación describiremos el objetivo de cada uno de estos procesos.

**Construcción de la representación.** El primer paso obligado es el *indexado* de las oraciones ( $S$ ), actividad que denota hacer el mapeo de una oración  $s_i$  en una forma compacta de su contenido. La representación más comúnmente utilizada para representar textos es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información. Esto es, cada texto  $s_i$  es representado como el vector  $\vec{s}_i = \langle w_{k_i}, \dots, w_{|\tau|_i} \rangle$ , donde  $\tau$  es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún elemento de  $S$ , mientras que  $w_{k_i}$  representa la importancia del término  $t_k$  dentro del contenido del documento  $s_i$ . Este método de representación, también conocido como bolsa de palabras (BoW), propone varios esquemas para definir  $w_{k_i}$ , en particular, para nuestro sistema desarrollado se utilizó el esquema de pesado TF-IDF [2].

**Clustering.** El *Clustering* o agrupamiento es un proceso que nos permite encontrar los distintos sub-temas contenidos en la colección de documentos. Este proceso nos ayudará a identificar la información relevante y redundante que existe en dicha colección de documentos. Para este proceso se utilizó el algoritmo de agrupamiento estrella [1] (Algoritmo 1). Entre las bondades de este método se tiene que induce de manera natural el número de grupos [6] en una colección. La salida del algoritmo son grupos de documentos en forma de estrella en donde el centro de la estrella es el documento más representativo del grupo y los satélites son los documentos que están relacionados a ese centro de estrella.



**Datos:** Grafo  $G$ , umbral de similitud  $\sigma$   
**Resultado:** Grupos en forma de estrella  
 Calcular  $G_\sigma = (V, E_\sigma)$  donde  $E_\sigma = \{e \in E : w(e) \geq \sigma\}$ ;  
 Poner cada vértice en  $G_\sigma$  inicialmente marcado como *no-visitado*;  
**mientras** *no estén todos los vértices marcados como visitados* **hacer**  
     1. Tomar el vértice de mayor grado que tenga la etiqueta “no-visitado”  
         como centro de la estrella;  
     2. Construir un grupo con éste como centro de la estrella y sus satélites con  
         sus vértices asociados;  
     3. Marcar cada nodo de la estrella recién construida como “visitados”;  
**fin**

**Algoritmo 1:** Algoritmo de agrupamiento estrella.

La entrada del algoritmo de agrupamiento estrella es un grafo  $G$  el cual es un grafo completo con aristas de peso variable. Para nuestro caso, este grafo es generado a través de considerar a todas las oraciones  $s_i$  de la colección de documentos de entrada como los vértices de  $G$ , mientras que las aristas llevan como peso el valor de similitud entre los vértices respectivos<sup>6</sup>. De esta forma, el grafo umbralizado  $G_\sigma$ , es un grafo no dirigido obtenido de  $G$  al ir eliminando todas las aristas cuyos pesos son menores a  $\sigma$ .

Observe que otro parámetro de entrada del método descrito en el algoritmo 1 es el valor de  $\sigma$ , el cual representa un valor de similitud mínimo que deben de tener los elementos de cada grupo formado. El valor de  $\sigma$  se define por medio de la similitud media entre los documentos de entrada más la desviación estándar. De esta forma, aseguramos que los grupos formados (sub-temas) sean realmente relacionados.

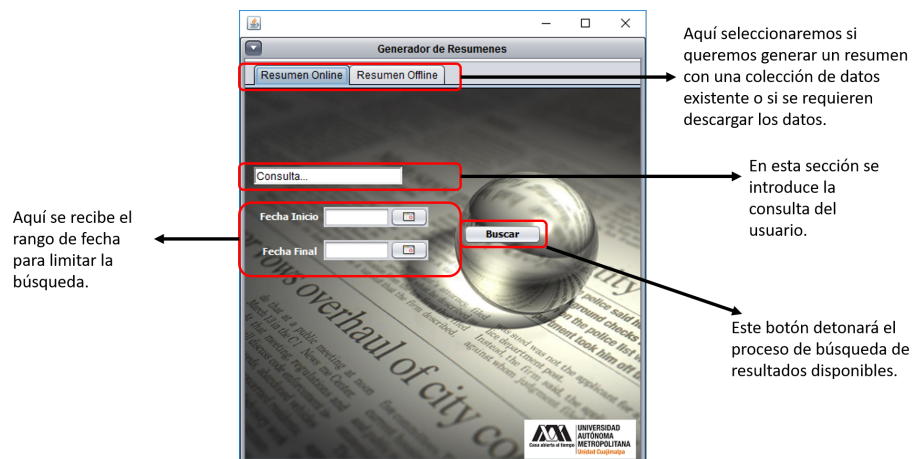
**Extracción de oraciones importantes.** El objetivo de este proceso es la construcción final del resumen. Para esto, se toma como entrada los diferentes sub-temas encontrados por la etapa de agrupamiento y la consulta del usuario que deberá guiar el resumen. Para esto se identifican las oraciones de mayor similitud de cada sub-tema con la consulta del usuario. De esta forma, estas oraciones se incorporan en el resumen hasta cubrir la tasa de compresión solicitada por el usuario.

## 4. Sistema desarrollado

El sistema desarrollado consiste en una aplicación de escritorio la cual permite al usuario generar resúmenes a partir de noticias descargadas del sitio ‘El Universal’, a lo cual le denominamos “Resumen en-línea”. Por otro lado, también se incorporó un modo en el cual el usuario puede proporcionar directamente la colección de documentos que quiere analizar, a lo cual denominamos “Resumen fuera-de-línea”. Para la programación de esta herramienta se utilizó los lenguajes

<sup>6</sup> Para la versión actual del sistema, solo la medida del coseno está considerada [2].

Python y Java, lo cual permite que el sistema funcione en plataformas tanto Windows como Linux<sup>7</sup>.



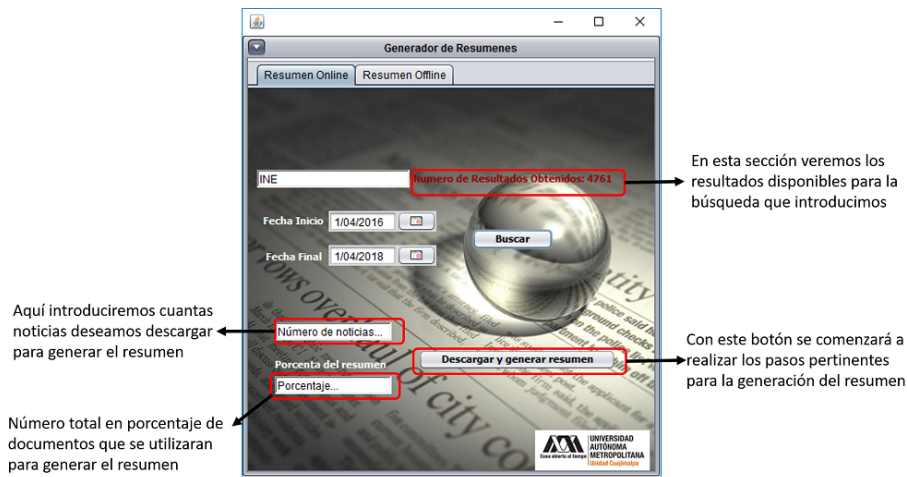
**Fig. 4.** Visualización de la interfaz en la sección “Resumen en-línea”. Esta ventana permite conectarse a Internet y descargar todos aquellos documentos relacionados con la consulta proporcionada entre los rangos de fechas especificados.

En la figura 4 se muestra la pantalla del modo “Resumen en-línea”. Como se puede ver en la figura los parámetros que se requieren del usuario en esta primera pantalla es una consulta y un rango de fechas entre los cuales desea buscar documentos. Una vez se hace la búsqueda, el sistema mostrará una ventana como la que se visualiza en la figura 5. En esta pantalla el usuario puede ver cuántos documentos se encontraron, y además de que se le da la opción de definir un número específico de documentos a descargar. Al mismo tiempo se le pide que defina una tasa de compresión para la generación del resumen final. Al hacer click en el botón ‘Descargar y generar resumen’, el proceso de construcción del resumen comenzará.

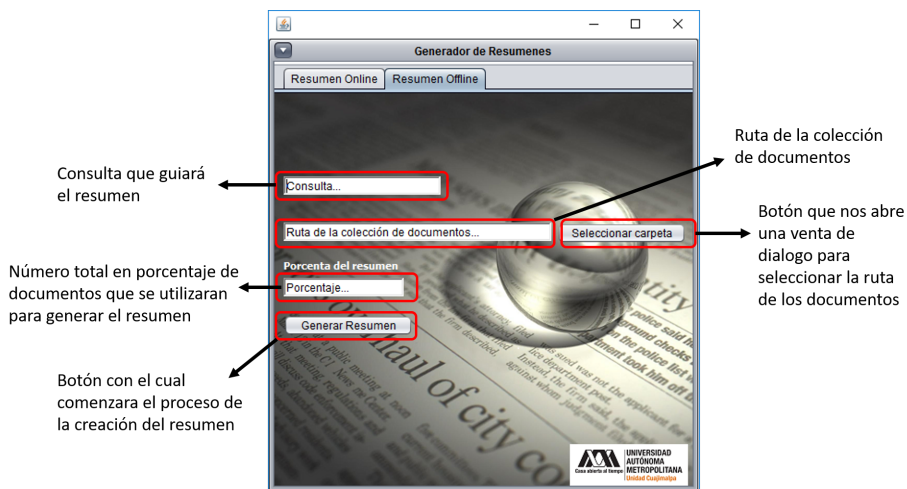
En la figura 6 se muestra el modo de operación ‘Resumen Off-line’. Bajo este modo de funcionamiento no es necesario tener una conexión a Internet para que el sistema trabaje, sin embargo el usuario deberá indicar la ubicación de los documentos que se quieren analizar. Continuando con el ejemplo anterior, si se quisiera seguir analizando una colección previamente descargada, basta con especificar la ruta en donde se guardaron los documentos.

Note que en el modo ‘fuera-de-línea’ se pide nuevamente una *consulta (query)* al usuario así como la tasa de compresión del resumen a construir. Bajo este esquema es posible generar resúmenes distintos dependiendo de la consulta proporcionada por el usuario.

<sup>7</sup> La versión ejecutable de la herramienta desarrollada se puede descargar desde: <http://ccd.cua.uam.mx/~evillatoro/Resources/SistemasGeneradordeResmenes.rar>



**Fig. 5.** Interfaz en la sección “Resumen en-línea” que se muestra una vez que se han identificado los documentos relevantes a la consulta. En rojo se muestran el total de documentos encontrados; una vez que el usuario define la tasa de compresión y presiona el botón de descarga, se comienza el proceso de generación del resumen.



**Fig. 6.** Ventana que se muestra en el modo de funcionamiento ‘fuera-de-línea’. El usuario debe especificar la ruta de dónde están los documentos sobre los cuales quiere trabajar, la consulta, y la tasa de compresión del resumen. Una vez definidos estos parámetros, se procede a la construcción del resumen.

Finalmente, una vez construido el resumen se visualizará una ventana como la mostrada en la figura 7 donde será posible ver el resumen construido. Este resumen es también almacenado en la ubicación donde se encuentra la aplicación

**Tabla 1.** Resúmenes generados ante dos consultas distintas empleando la misma colección de documentos.

<p><i>consulta<sub>1</sub></i> : <i>Dinero invertido en armas y la relación del narcotráfico con Estados Unidos.</i></p> <p><i>Resumen.1:</i> De acuerdo con Pamela Starr, experta en la relación entre Estados Unidos y México de la Universidad del Sur de California, la medida tiene sentido y guarda relación con la supuesta oferta que Trump hizo al presidente mexicano, Enrique Peña Nieto, de colaborar en la lucha contra el narcotráfico y los cárteles de la droga, la cual se habría planteado durante la conversación telefónica que mantuvieron hace un par de semanas. Según expedientes judiciales, la organización efectuaba sus actividades desde la zona suburbana de Dallas, y Treviño Morales había invertido 16 millones de dólares de dinero proveniente del narcotráfico en la compra y entrenamiento de caballos para que participaran en carreras en el suroeste de Estados Unidos. Desde 2004, Yarrington enfrenta acusaciones por presunto narcotráfico y lavado de dinero. El narcotráfico, la inseguridad.</p>
<p><i>consulta<sub>2</sub></i> : <i>Los estados con mayor índice de violencia y corrupción policiaca con el narcotráfico.</i></p> <p><i>Resumen.2:</i> Aseguró que las fuerzas de seguridad están siendo maltratadas, hay corrupción en los mandos, convenios con los cárteles y corrupción en los ministerios públicos. Evitar que fueran reclutados por los criminales, con la prevención social de la violencia y la delincuencia, y por el otro, asegurar mayores oportunidades de educación de calidad para las y los jóvenes. El narcotráfico, la inseguridad. .<sup>En</sup> 20 años de gobierno del PRD y Morena, nuestras familias han visto como se deteriora la ciudad por la corrupción.</p>

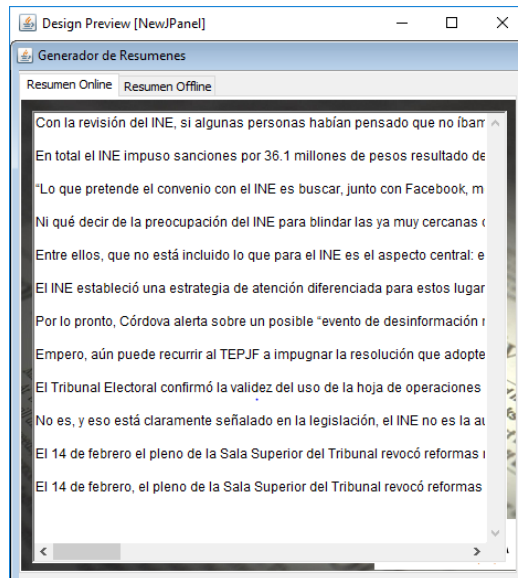
de escritorio del sistema.

En la Tabla 1 se muestra un ejemplo de dos resúmenes contruidos a partir de necesidades de información diferentes empleando la misma colección de documentos. Para el ejemplo mostrado se descargaron 3,883 noticias empleando la consulta *Narcotráfico*, mientras que para la generación de los resúmenes se emplearon las siguientes consultas: *Dinero invertido en armas y la relación del narcotráfico con Estados Unidos*, y *Los estados con mayor índice de violencia y corrupción policiaca con el narcotráfico*. Para la construcción del resumen se solicitó al sistema una tasa de compresión del 1 %.

Como es posible observar, el sistema genera resúmenes muy diferentes a necesidades de información planteadas, lo cual permite, hasta cierto punto, responder a las necesidades de usuarios diferentes. Internamente, una vez que el sistema identifica a las oraciones candidatas para estar en el resumen, las mismas son ordenadas de acuerdo a su nivel de similitud con la consulta, y son colocadas en este orden hasta cubrir la restricción de tamaño solicitado.

## 5. Conclusiones

Este trabajo describe la implementación de un sistema de generación de resúmenes de múltiples documentos guiado por consulta. El sistema desarrollado



**Fig. 7.** Ventana que se muestra con el resumen final. Una vez finalizado el proceso de generación del resumen, se muestra una ventana de texto con las piezas de información que se identificaron como más relevantes. Si el usuario desea generar otro resumen con otra consulta diferente, bastará con volver a las ventanas previas.

es capaz de responder a distintas necesidades de información por medio de considerar una consulta proporcionada en lenguaje natural por uno o varios usuarios. Una de las ventajas del sistema desarrollado es que se incorporó un módulo de recuperación de información, el cual se conecta en tiempo real a el sitio de noticias de El Universal para descargar documentos que satisfacen una necesidad de información inicial. Posteriormente, usuarios con diferentes perfiles o necesidades de información pueden generar variados resúmenes que responden a consultas específicas.

Para el desarrollo del presente sistema se utilizaron técnicas de Inteligencia Artificial así como de Procesamiento de Lenguaje Natural. Por un lado, en lo que respecta a PLN, técnicas tradicionales de representación de textos así como de cálculo de similitud entre documentos son utilizadas para procesar los documentos. Por otro lado, técnicas de aprendizaje no supervisado son empleadas para la identificación de tópicos importantes entre los documentos descargados.

Durante la implementación del sistema fue posible observar que mejores formas de representación de la información pueden ser incorporadas al sistema, así como técnicas más eficientes de agrupamiento. Como parte del trabajo futuro, queremos incorporar representaciones que capturen de manera más eficiente la semántica de los documentos, de forma que sea posible identificar oraciones relevantes aunque éstas no compartan términos de manera explícita. Agregado a eso, queremos emplear estrategias de agrupamiento jerárquico, técnica que

permitirá construir el resumen considerando otro esquema de organización de la información relevante. Ambas estrategias permitirán la construcción de resúmenes más valiosos para el usuario final.

**Agradecimientos.** Agradecemos a la Coordinación de la Licenciatura Tecnologías y Sistemas de Información de la Universidad Autónoma Metropolitana, Unidad Cuajimalpa por el apoyo otorgado para la realización de este trabajo.

## Referencias

1. Aslam, J., Pelekhev, K., Rus, D.: A practical clustering algorithm for static and dynamic information organization. In: Proceedings of the 1999 Symposium on Discrete Algorithms. pp. 208–217 (1999)
2. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern information retrieval, vol. 463. ACM press New York (1999)
3. Burges, C.J., Ragno, R., Le, Q.V.: Learning to rank with nonsmooth cost functions. In: Advances in neural information processing systems. pp. 193–200 (2007)
4. Canhasi, E., Kononenko, I.: Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. Expert Systems with Applications 41(2), 535–543 (2014)
5. Chuang, W.T., Yang, J.: Text summarization by sentence segment extraction using machine learning algorithms. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 454–457. Springer (2000)
6. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) 31(3), 264–323 (1999)
7. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of research and development 2(2), 159–165 (1958)
8. Luna-Tlatelpa, L., Villatoro-Tello, E., Ramírez-de-la Rosa, G., Rivero-Moreno, C.J.: Resúmenes de múltiples documentos guiados por consulta empleando representaciones distribucionales. Research in Computing Science 134, 127–139 (2017)
9. Ouyang, Y., Li, W., Li, S., Lu, Q.: Applying regression models to query-focused multi-document summarization. Information Processing & Management 47(2), 227–237 (2011)
10. Petsche, T., Mozer, M.C., Jordan, M.I.: Advances in Neural Information Processing Systems 9: Proceedings of the 1996 Conference. MIT Press (1997)
11. Wang, L., Raghavan, H., Castelli, V., Florian, R., Cardie, C.: A sentence compression based framework to query-focused multi-document summarization. arXiv preprint arXiv:1606.07548 (2016)

## Identificación de etiquetas semánticas para su uso en diálogos

Andrés Vázquez, David Pinto, Darnes Vilariño

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
México

{andrex,dpinto,darnes}@cs.buap.mx

**Resumen.** En la interacción humano-robot la generación automática de diálogos es una parte importante. Los diálogos generados deben garantizar una conversación coherente entre el humano y el robot, se espera que la interacción sea lo más natural y eficaz que se pueda. En este trabajo se plantea el uso de entidades semánticas como elementos básicos que posteriormente serán utilizados para la realización de un módulo de comprensión del lenguaje como parte de un sistema de diálogos. Para poder identificar las entidades semánticas en los diálogos se evaluaron dos herramientas para reconocer entidades nombradas (NER), que utilizan como núcleo un clasificador basado en Campos Aleatorios Condicionales. Los resultados que se obtienen en este trabajo permiten afirmar que estos dos NER tienen buenos resultados en el reconocimiento de las entidades semánticas (tokens) pues de manera global superan el 84% de exactitud. El corpus utilizado en este trabajo es el corpus español DIHANA, el cual está compuesto de diálogos sobre un sistema de información de consultas telefónicas sobre horarios y precios de trenes de largo recorrido.

**Palabras clave:** sistema de diálogos, entidad semántica, reconocedor de entidad nombrada.

## Identification of Semantic Tags for Use in Dialogues

**Abstract.** In the human-robot interaction the automatic generation of dialogues is an important part, the generated dialogues must guarantee a coherent conversation between the human and the robot, the interaction is expected to be as natural and effective as possible. In this paper, the use of semantic entities as basic elements is proposed, which will later be used for the realization of a language understanding module as part of a dialog system. To identify the semantic entities in the dialogues, two tools for recognizing named entities (NER) were evaluated, using as a nucleus a classifier based on Conditional Random Fields. The results obtained in this work allow us to affirm that these two NERs have satisfactory results in the recognition of semantic entities (tokens) because, overall, they exceed 84% accuracy. The corpus used in this

work is the Spanish DIHANA corpus, which is composed of dialogues about an information system of telephone queries about schedules and prices of long-distance trains.

**Keywords:** dialog system, semantic entity, named entity recognizer.

## 1. Introducción

A medida que mejora la comprensión del lenguaje y la tecnología de generación automática de diálogos, también aumenta el interés en la construcción de sistemas de conversación de usuario, que pueden ser utilizados para una variedad de aplicaciones tales como planificación de viajes, sistemas tutoriales o soporte técnico basado en chat. La descripción de algunos de estos sistemas desarrollados en los últimos años se puede encontrar en [1–8].

Los sistemas de diálogo hablado o sistemas conversacionales son una tecnología concebida para facilitar la interacción natural mediante el habla, entre una persona y una computadora, son una interfaz hombre-máquina capaz de reconocer y comprender una entrada hablada y reproducir una salida oral como respuesta. Los sistemas de diálogo consisten de una estructura modular en la que cada módulo se ocupa de determinadas tareas en interacción con todos los demás módulos, normalmente los módulos que conforman a un sistema de diálogo son: el reconocedor del habla, el módulo de comprensión del lenguaje, el gestor del diálogo y el módulo de generación de respuesta.

En este trabajo nos enfocamos únicamente al módulo de comprensión del lenguaje, en donde se maneja el concepto de entidad semántica y cómo reconocer estas entidades semánticas a través del uso de reconocedores de entidades nombradas (Named Entity Recognizer, NER) [9] y posteriormente sea más fácil el proceso de comprensión del lenguaje en la interpretación semántica (secuencia de unidades semánticas) y que en un futuro forme parte de un sistema de diálogo del tipo pregunta-respuesta.

El presente trabajo está estructurado de la siguiente manera: en la sección 2 se describe el corpus que se utilizó para hacer la prueba de los NER; en la sección 3 se describe el proceso general de un NER y se presentan los resultados obtenidos de dos herramientas NER utilizadas en este trabajo; finalmente, en la sección 4 se presentan las conclusiones y el trabajo a futuro.

## 2. Estudio de caso corpus DIHANA

Con la finalidad de evaluar la identificación de entidades semánticas en el uso de diálogos se utilizó el corpus DIHANA [10]. El corpus en español DIHANA está compuesto por 900 diálogos sobre un sistema de información de consultas telefónicas sobre horarios y precios de trenes de largo recorrido. Fue adquirido por 225 hablantes



**Tabla 1.** Ejemplo de algunos diálogos del corpus DIHANA.

No.	Diálogo
U0000	hola buenos días mira quería saber horario de trenes para ir a cuenca
U0001	sí que quería saber horarios de trenes para ir a cuenca
U0002	pues quiero salir el día treinta de junio
U0003	pues no gracias
U0004	hola buenos días quería saber horarios para ir a Barcelona
U0005	pues quiero salir el treinta de julio
U0006	pues que quiero ir el treinta de julio
U0007	sí efectivamente y quisiera ir en un tren que fuese rápido tengo que estar allí antes de las ocho de la tarde vamos
U0008	quisiera saber el horario de un tren que llegue allí antes de las ocho

U0000:hola buenos días mira quería saber horario de trenes para ir a cuenca	
hola buenos días:	<b>cortesía</b>
mira quería saber:	<b>consulta</b>
horario de trenes para ir:	<b>&lt;hora&gt;</b>
a cuenca:	<b>ciudad_destino</b>
U0001:sí que quería saber horarios de trenes para ir a cuenca	
sí:	<b>&lt;afirmacion&gt;</b>
que:	<b>nada</b>
quería saber:	<b>consulta</b>
horarios de trenes para ir:	<b>&lt;hora&gt;</b>
a cuenca:	<b>ciudad_destino</b>

**Fig. 1.** Dos diálogos y sus correspondientes entidades semánticas.

diferentes (153 hombres y 72 mujeres). Hay 6,280 turnos de usuario y 9,133 turnos del sistema. El tamaño del vocabulario es de 823 palabras. La cantidad total de señal de voz fue de aproximadamente cinco horas y media (véase la tabla 1).

La Fig. 1 muestra dos diálogos y sus correspondientes entidades semánticas.

La adquisición del corpus DIHANA se llevó a cabo por medio de un prototipo inicial, utilizando la técnica del Mago de Oz (WoZ). Esta adquisición solo se restringió a nivel semántico (es decir, los diálogos adquiridos están relacionados con un dominio de tareas específico) y no se restringió a nivel léxico y sintáctico (habla espontánea). En este proceso de adquisición, el control semántico fue proporcionado por la definición de escenarios que el usuario tenía que cumplir y por la estrategia WoZ, que define el comportamiento del sistema de adquisición.

### 3. Evaluación de reconocedores de entidades nombradas (NER)

En este trabajo se emplean dos reconocedores de entidades nombradas (NER) para reconocer entidades semánticas en los diálogos del corpus DIHANA y esto nos ayude en el proceso de comprensión del lenguaje en la interpretación semántica (secuencia de unidades semánticas) de los diálogos.

### 3.1. Descripción general del proceso de un NER

En el procesamiento de lenguaje natural, el reconocimiento de entidad nombrada es una tarea de extracción de información que busca ubicar y clasificar elementos en texto en categorías predefinidas, tales como personas, organizaciones, lugares, expresiones de tiempo y cantidades entre otros. Por ejemplo, en el siguiente texto [9]:

“Jim compró 300 acciones de Acme Corp. en 2006”

al aplicarle un NER se obtiene que se reconocen las entidades persona, organización y tiempo:

“[Jim] (persona) compró 300 acciones de [Acme Corp.] (organización) en [2006] (tiempo).”

El proceso de un NER normalmente consta de tres etapas:

**Primera etapa.** Corresponde a la preparación de los datos de entrenamiento, la cual se describe gráficamente en la Fig. 2, en donde inicialmente se tiene un corpus identificado como un “Archivo plano”, al cual se le aplica un método de “tokenización” y posteriormente se realiza el “etiquetado” del corpus de acuerdo con los requerimientos de cada NER.

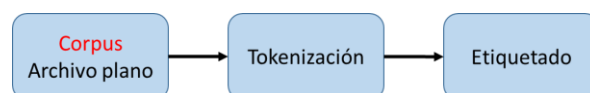


Fig. 2. Preparación de los datos de entrenamiento para un NER.

**Segunda etapa.** Esta etapa corresponde a la etapa de entrenamiento del NER, en donde se utiliza parte del corpus generado en la primera etapa (Training) y como resultado se obtiene un modelo de NER entrenado, el cual se describe en la Fig. 3.

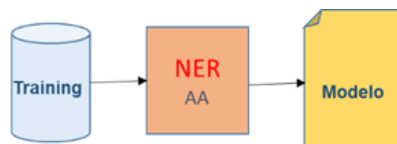


Fig. 3. Etapa de entrenamiento de un NER.

**Tercera etapa.** Corresponde a la evaluación del modelo obtenido en la etapa anterior y es aquí donde se ve el desempeño del modelo, verificando la precisión de las entidades semánticas identificadas, como se muestra en la Fig. 4.

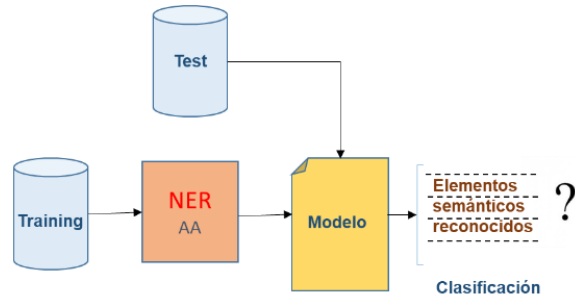


Fig. 4. Etapa de evaluación de un NER.

Tabla 2. Descripción de los archivos de entrenamiento y prueba.

Característica	Archivo de entrenamiento	Archivo de prueba
Intervenciones	6279	4878
Etiquetas	30	28

Tabla 3. Precisión obtenida en la identificación de tokens.

NER	No. Tokens	Precisión
Stanford	10,485	0.84
Xingdi (Eric) Yuan	10,485	0.89

### 3.2. Resultados experimentales

Las dos herramientas NER que se utilizaron en este trabajo fueron: el NER de Stanford [11] y el NER de Xingdi (Eric) Yuan [12]. Estas dos herramientas utilizan el modelo de secuencia de campos aleatorios condicionales (CRF) para la clasificación.

En ambos modelos se utilizaron los mismos archivos de entrenamiento y de prueba del corpus DIHANA, los cuales se describen en la Tabla 2.

Las intervenciones en cada archivo son diferentes, así como las etiquetas que se describen son etiquetas diferentes que pueden aparecer más de una vez en los archivos de entrenamiento y de prueba.

Los resultados globales de la precisión obtenida en los dos NER se muestran en la Tabla 3, aquí la precisión es el número de tokens de un tipo dado que el modelo identificó correctamente, entre el número total de tokens que el modelo predijo que era de ese tipo.

Si bien, se puede observar que el etiquetador de entidades nombradas Xingdi (Eric) Yuan obtiene el mejor resultado global, es muy importante analizar el resultado obtenido para cada una de las clases semánticas clasificadas. Así, en la Tabla 4, se han incluido dichos resultados, los cuales procedemos a analizar a continuación.

CRFClassifier tagged 10485 words in 1 documents at 141.85 words per second.

Entity	P	R	F1	TP	FP	FN
<afirmacion>	0.9730	0.9665	0.9698	433	12	15
<duracion>	0.0000	0.0000	0.0000	0	0	2
<hora>	0.8152	0.8113	0.8132	172	39	40
<hora_llegada>	0.9412	0.9412	0.9412	16	1	1
<hora_salida>	0.8571	0.6667	0.7500	6	1	3
<negacion>	0.8561	0.9154	0.8848	238	40	22
<no_entendido>	0.5000	0.1429	0.2222	1	1	6
<precio>	0.7429	0.7280	0.7354	182	63	68
<tipo_tren>	0.5833	0.7000	0.6364	7	5	3
albacete	0.0000	1.0000	0.0000	0	1	0
barcelona	1.0000	1.0000	1.0000	2	0	0
ciudad	0.7778	0.6563	0.7119	21	6	11
ciudad_destino	0.9789	0.9754	0.9772	278	6	7
ciudad_origen	0.9872	0.9625	0.9747	154	2	6
clase_billete	0.7000	0.6364	0.6667	21	9	12
coletilla	0.7612	0.7846	0.7727	102	32	28
consulta	0.8003	0.8082	0.8042	573	143	136
cortesía	0.8537	0.8140	0.8333	105	18	24
cÃ;ceres	0.0000	0.0000	0.0000	0	0	1
cÃ;diz	0.0000	0.0000	0.0000	0	0	1
fecha	0.9263	0.9337	0.9300	352	28	25
gasteiz	0.0000	0.0000	0.0000	0	0	1
granada	0.0000	1.0000	0.0000	0	2	0
hora	0.8750	0.8537	0.8642	175	25	30

Fig. 5. Inicio de la ejecución del NER Stanford.

ciudad_destino	0.9789	0.9754	0.9772	278	6	7
ciudad_origen	0.9872	0.9625	0.9747	154	2	6
clase_billete	0.7000	0.6364	0.6667	21	9	12
coletilla	0.7612	0.7846	0.7727	102	32	28
consulta	0.8003	0.8082	0.8042	573	143	136
cortesía	0.8537	0.8140	0.8333	105	18	24
cÃ;ceres	0.0000	0.0000	0.0000	0	0	1
cÃ;diz	0.0000	0.0000	0.0000	0	0	1
fecha	0.9263	0.9337	0.9300	352	28	25
gasteiz	0.0000	0.0000	0.0000	0	0	1
granada	0.0000	1.0000	0.0000	0	2	0
hora	0.8750	0.8537	0.8642	175	25	30
la_coruña	1.0000	0.5000	0.6667	1	0	1
lleida	0.0000	0.0000	0.0000	0	0	1
logroño	0.0000	1.0000	0.0000	0	1	0
m_llegada	0.6962	0.7051	0.7006	55	24	23
m_salida	0.8492	0.8492	0.8492	107	19	19
madrid	0.0000	1.0000	0.0000	0	1	0
mañana	0.0000	0.0000	0.0000	0	2	1
no	1.0000	1.0000	1.0000	83	0	0
nombre_atributo	0.0000	0.0000	0.0000	0	0	1
not	0.0000	0.0000	0.0000	0	2	9
numero_relativo_orden	0.6250	0.7143	0.6667	5	3	2
pamplona	0.0000	0.0000	0.0000	0	0	1
precio	0.6667	0.4000	0.5000	2	1	3
pues	0.0000	0.0000	0.0000	0	0	1
san_sebastián	0.0000	1.0000	0.0000	0	1	0
sÃ-	1.0000	1.0000	1.0000	131	0	0
tarragona	1.0000	0.5000	0.6667	1	0	1
tipo_tren	0.6458	0.6596	0.6526	62	34	32
tipo_viaje	0.5067	0.5135	0.5101	76	74	72
valladolid	0.0000	0.0000	0.0000	0	0	1
viernes	0.0000	0.0000	0.0000	0	0	1
Totals	0.8494	0.8462	0.8478	3361	596	611

andrex@arcturus:~/stanford-ner-2017-06-09\$

Fig. 6. Final de la ejecución del NER Stanford.

**Tabla 4.** Resultados obtenidos de la evaluación de los dos NER.

Etiquetas semánticas	Frecuencia Training	Frecuencia Test	Exactitud NER Stanford	Exactitud NER Xingdi (Eric) Yuan
<hora_llegada>	73	73	94.12%	100.00%
fecha	1678	1655	92.63%	98.63%
ciudad_destino	611	601	97.89%	98.36%
<afirmacion>	541	527	97.30%	97.41%
hora	978	949	87.50%	97.03%
ciudad_origen	323	310	98.72%	95.98%
<precio>	917	866	74.29%	94.44%
<hora>	618	578	81.52%	93.53%
consulta	1935	1758	80.03%	90.85%
<negacion>	344	302	85.61%	87.79%
numero_relativo_orden	24	21	62.50%	87.50%
m_salida	186	159	84.92%	85.48%
cortesía	253	215	85.37%	84.98%
m_llegada	136	115	69.62%	84.56%
coletilla	225	188	76.12%	83.56%
tipo_viaje	395	328	50.67%	83.04%
<tipo_tren>	65	52	58.33%	80.00%
tipo_tren	246	187	64.58%	76.02%
clase_billete	94	70	70.00%	74.47%
ciudad	44	30	77.78%	68.18%
<hora_salida>	38	20	85.71%	52.63%
precio	15	6	66.67%	40.00%
O	471	171	34.15%	36.31%
not	12	2	0.00%	16.67%
<duracion>	8	0	0.00%	0.00%
<no_entendido>	22	0	0.00%	0.00%
nombre_atributo	4	0	0.00%	0.00%

Las etiquetas semánticas “<hora\_llegada>”, “fecha”, “ciudad\_destino”, “<afirmación>”, “hora” y “ciudad\_origen” son las que han obtenido el mayor grado de exactitud durante el proceso de identificación automática con un valor que supera el 95%.

Consideramos que este resultado se encuentra fundamentado en dos cosas: primeramente, en la cantidad de datos usados durante la fase de entrenamiento, pero también a la inherente naturaleza de las palabras del lenguaje natural asociadas a dichas entidades semánticas, pues existe cierta uniformidad para expresar, horarios y fechas, y en cuanto a los nombres de las ciudades, se considera que el número es ciertamente limitado, lo cual facilita el proceso de identificación.

Aun así, existen diferencias importantes entre los dos NER evaluados, que pueden ser vistas a detalle en la Tabla 4. La conclusión obtenida es que el NER Xingdi (Eric) Yuan es más estable que el proporcionado por Stanford.

#### 4. Conclusión y trabajo futuro

El objetivo principal de este trabajo fue el de evaluar la tarea de comprensión del lenguaje, asociada a la generación de diálogos en lenguaje natural. Hemos utilizado el concepto de reconocimiento de entidades nombradas para hacer frente al problema de traducción de sentencias del lenguaje natural a entidades semánticas. Se evaluaron dos sistemas que utilizan como núcleo un clasificador basado en Campos Aleatorios Condicionales. Los resultados que se obtienen en el presente trabajo permiten afirmar que estos dos NER tienen buenos resultados en el reconocimiento de las entidades semánticas (tokens) pues de manera global superan el 84% de exactitud. En particular, es el NER Xingdi (Eric) Yuan el que produce los resultados más confiables y estables, por lo que consideramos importante utilizarlo próximamente durante la fase de comprensión del lenguaje.

Como trabajo futuro se tiene contemplado tomar las entidades semánticas reconocidas y representar los diálogos mediante inferencia gramatical con la finalidad de inducir una gramática de los componentes semánticos que apoye en la generación de gramáticas más complejas que incluyan el concepto de pregunta-respuesta y empatamiento de contenido con su correspondiente generación de respuesta validada.

También se tiene contemplado probar otros dos NER con el enfoque de redes neuronales recurrentes (LSTM) y convolucionales con la finalidad de verificar si su rendimiento es mejor que aquellos basados en campos aleatorios condicionales.

**Agradecimientos.** Este trabajo de investigación ha sido parcialmente respaldado por la beca CONACyT # 80286, bajo el Programa de Doctorado en Ingeniería del Lenguaje y del Conocimiento (LKE) de la Benemérita Universidad Autónoma de Puebla.

#### Referencias

1. Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington L.: JUPITER: A Telephone-Based Conversational Interface for Weather Information. *IEEE Transactions on Speech and Audio Processing*, 8(1) (2000)
2. Glass, J., Weinstein, E.: Speech builder: facilitating spoken dialog system development. In: *Proceedings of EUROSPEECH*, pp. 1335–1338 (2001)
3. Córdoba, R., San-Segundo, R., Montero, J.M., Colás, J., Ferreiros, J. Macias-Guarasa, J., Pardo, J.M.: An interactive directory assistance service for Spanish with large vocabulary recognition. In: *Proceedings of EUROSPEECH*, pp. 1279–1282 (2001)
4. Hurtado, L., Blat, F., García, F., Grau, S., Griol, D., Sanchis, E., Segarra, E., Torres, F.: Sistema de diálogo para el Proyecto DIHANA. *Procesamiento del Lenguaje Natural*, 35, pp. 453–454 (2005)
5. Griol, D., Hurtado, L., Sanchis, M., Segarra, E.: Dos aproximaciones basadas en reglas para la gestión del diálogo. *Procesamiento del Lenguaje Natural*, 35, pp. 213–220 (2005)
6. Pietquin, O.: Inverse Reinforcement Learning for Interactive Systems. *ACM (MLIS'13)* (2013)
7. Lison, P.: Model-based Bayesian Reinforcement Learning for Dialogue Management (2013)

8. Barlier, M., Perolat, J., Laroche, R., Pietquin, O.: Human-Machine Dialogue as a Stochastic Game. In: Proceedings of the SIGDIAL 2015 Conference, Prague, Czech Republic. pp. 2–11 (2015)
9. WIKIPEDIA: [https://en.wikipedia.org/wiki/Named-entity\\_recognition](https://en.wikipedia.org/wiki/Named-entity_recognition) (2018)
10. Benedí, J., Lleida, E., Varona, A., Castro, M., Galiano, I., Justo, R., López de Letona, I., Miguel, A.: Design and acquisition of a telephone spontaneous speech dialogue corpus in spanish: Dihana. In: Fifth International Conference on Language Resources and Evaluation (LREC), pp. 1636–1639 (2006)
11. NER Stanford: <https://nlp.stanford.edu/software/CRF-NER.html> (2018)
12. NER Xingdi: <https://github.com/xingdi-eric-yuan/nerpp> (2018)





# Representaciones vectoriales de palabras de un corpus de normas de asociación

Jorge Reyes Magaña<sup>1</sup>, Helena Gómez Adorno<sup>2</sup>,  
Gemma Bel Enguix<sup>2</sup>, Gerardo Sierra<sup>2</sup>

<sup>1</sup> Universidad Autónoma de Yucatán,  
Facultad de Matemáticas, Mérida, Yucatán,  
México

<sup>2</sup> Universidad Nacional Autónoma de México,  
Instituto de Ingeniería, Ciudad de México,  
México

jorge.reyes@correo.uady.mx,  
{hgomez, gbele, gsierra}@iingen.unam.mx

**Resumen.** Las representaciones vectoriales de palabras son muy eficientes para muchas tareas de procesamiento del lenguaje natural y para su construcción es necesario entrenarlos con una gran cantidad de datos para obtener vectores de buena calidad que permitan realizar las tareas con éxito. En este trabajo, presentamos un método basado en el algoritmo *node2vec* para aprender vectores de palabras usando un grafo que ha sido construido tomando como base un corpus de normas de asociación de palabras en español; el grafo construido contiene en los nodos las palabras y en las aristas diferentes pesos como son frecuencia, tiempo y fuerza de asociación. Los recursos computacionales utilizados por este método son razonables y asequibles. Esto nos permite obtener vectores de palabras de buena calidad incluso desde un corpus pequeño. Evaluamos nuestro método en un corpus de similitud y relacionalidad de palabras, obteniendo resultados comparables a los obtenidos con *word2vec* entrenados en un corpus de mil millones de palabras.

**Palabras clave:** vectores de palabras, normas de asociación de palabras.

## Word Embeddings Learned on Word Association Norms

**Abstract.** Word embeddings are powerful tools for many natural language processing tasks. In order to obtain embeddings useful for different tasks, it is necessary a large training corpus. In this work, we present a method based on the *node2vec* algorithm to learn word embeddings from a graph built using a corpus of word association norm in Spanish. The nodes of the graph correspond to the words in the corpus, whereas the

edges are weighted with the frequency, time and association strength of a pair of words. The computational resources used by this technique is reasonable and affordable. This allows us to obtain good quality word embeddings even from a small corpus. We evaluated our word vectors in a word similarity and relatedness benchmark, achieving comparable results to those obtained with *word2vec* trained on a billion word corpus.

**Keywords:** word vectors, word association norms.

## 1. Introducción

La representación semántica de palabras en un espacio vectorial es un área de investigación muy activa en las últimas décadas. Modelos computacionales como la descomposición de valores singulares (SVD) y el análisis semántico latente (LSA) son capaces de modelar representaciones continuas de palabras (*word embeddings*) a partir de matrices término-documento. Ambos métodos pueden reducir un conjunto de datos de  $N$  dimensiones utilizando solo las dimensiones más importantes. Recientemente, Mikolov *et al.* [15] introdujeron *word2vec*, inspirado en la hipótesis distribucional que establece que las palabras en contextos similares tienden a tener significados similares [17]. Dicho método utiliza una red neuronal para aprender representaciones vectoriales de palabras al predecir otras palabras en su contexto. La representación vectorial de la palabra obtenida mediante *word2vec* tiene la asombrosa capacidad de preservar las regularidades lineales entre palabras.

Para construir un modelo de espacio vectorial adecuado y confiable, capaz de capturar la similitud semántica y las regularidades lineales de las palabras, se necesitan grandes volúmenes de texto. Aunque *word2vec* es rápido y eficiente de entrenar, y los vectores pre-entrenados generalmente están disponibles en línea, todavía es computacionalmente costoso procesar mediante este método grandes volúmenes de datos en entornos no comerciales, es decir, en computadoras personales.

La asociación libre es una técnica experimental comúnmente utilizada para descubrir la forma en que la mente humana estructura el conocimiento [6]. En las pruebas de asociación libre, se le pide a una persona que diga la primera palabra que se le viene a la mente en respuesta a una palabra *estímulo* dada. Mediante estos experimentos se obtienen unas compilaciones de relaciones léxicas, llamados Normas de Asociación de Palabras (NAP), que pueden reflejar tanto contenidos semánticos como episódicos [4].

El objetivo de este trabajo es presentar un método para aprender representaciones vectoriales de palabras a partir de nodos de un grafo obtenido a partir de un corpus NAP. Nuestra hipótesis es que los vectores aprendidos de este grafo mapean los contenidos de memoria semántica y episódica en el espacio vectorial, y así aprenden mejores representaciones. Grover y Leskovec [9] introdujeron un algoritmo llamado *node2vec* que es capaz de aprender mapeos de nodos a un espacio vectorial continuo teniendo en cuenta las vecindades de la red de los nodos. El algoritmo realiza caminos aleatorios sesgados para explorar diferentes

vecindarios con el fin de capturar no solo los roles estructurales de los nodos en la red, sino también las comunidades a las que pertenecen.

El presente trabajo está organizado de la siguiente manera. En la sección 2, discutimos el trabajo relacionado. En la Sección 3, presentamos el Corpus de Normas de la Asociación de Palabras para el Español Mexicano (NAP). En la sección 4, describimos el marco metodológico para aprender vectores de palabras a partir del NAP. La sección 5, muestra la evaluación de los vectores generados, utilizando como corpus de evaluación conjuntos de datos que contienen similitud de palabras en español. Finalmente, en la sección 6 sacamos algunas conclusiones y señalamos las posibles direcciones del trabajo futuro.

## 2. Trabajo relacionado

*Sinopalnikova y Smrz* [19] presentaron un marco metodológico para construir y extender redes semánticas con tesauros de asociaciones de palabras (WAT, por sus siglas en inglés). Además, hacen una comparación de la calidad y la información que ofrece WAT vs. otros recursos lingüísticos. Finalmente, los autores muestran que el WAT es comparable y se puede usar como un corpus balanceado de texto en ausencia de este.

*Borge-Holthoefer y Arenas* [4] describen un modelo para extraer relaciones de similitud semántica desde información de asociaciones libres (denominado RIM por sus sigla en inglés). Los autores aplican un método basado en redes para descubrir vectores de características en una red de asociaciones libres. Los vectores obtenidos fueron comparados con representaciones vectoriales basadas en LSA y el modelo WAS (Word Association Space). Los resultados de este trabajo indican que RIM puede extraer con éxito vectores de características de palabras desde una red de asociaciones libres.

En los últimos años, *Bel-Enguix et al.* [3] usaron técnicas de análisis de grafos para calcular asociaciones desde grandes colecciones de textos. Por otra parte, *Garimella et al.* [8] presentaron un modelo de asociaciones de palabras sensible al contexto demográfico basado en una arquitectura de redes neuronales con  $n$ -gramas no consecutivos. Este método mejoró el funcionamiento de las técnicas genéricas para calcular asociaciones que no tienen en cuenta la demografía del escritor.

En este trabajo se propone el uso de un recurso que recoge normas de asociación de palabras en español de México [1]. Desde este corpus, se aprenden representaciones vectoriales de las palabras.

## 3. Normas de asociación de palabras (NAP)

Las normas de asociación de palabras (WAN, por sus siglas en inglés), son corpus de asociaciones libres de palabras. Uno de los primeros ejemplos de estas recopilaciones es el que ofrecen *Kent y Rosanoff* [13], quienes usaron el método para estudiar la demencia a partir de 100 palabras estímulo emocionalmente neutras. Los autores llevaron a cabo el primer estudio a larga escala, con 1000

informantes, y concluyeron que existía uniformidad en la organización de las asociaciones, de manera que los adultos sanos comparten redes estables de conexiones de palabras [11].

Muchas lenguas cuentan con recopilaciones de Normas de Asociación de Palabras. En las décadas pasadas se han elaborado algunos trabajos interesantes con gran cantidad de voluntarios. Entre los recursos más conocidos en inglés accesibles desde la web se encuentra el *Edinburgh Associative Thesaurus*<sup>3</sup> (EAT) [14] y la compilación de *Nelson et al.* [16]<sup>4</sup>.

Para el español, existen algunos corpus de asociaciones libres, entre los que se encuentra el *Corpus de Normas de Asociación de Palabras para el Español de México* (NAP, a partir de ahora) [1], que es el primer recurso de este tipo especialmente recopilado entre hablantes nativos mexicanos del español. El corpus NAP fue elaborado con una muestra de 578 adultos jóvenes, hombres (239) y mujeres (339), con un rango de edad que va desde los 18 a los 28 años, y con un rango de educación de al menos 11 años. El número total de tokens del corpus es 65731, con 4704 palabras diferentes.

Para esta tarea se usaron 234 palabras estímulo, todas ellas sustantivos comunes tomados del *Inventario de Compresión y Producción de palabras MacArthur* [12] de *Jackson-Maldonado et al.* Es importante mencionar que si bien los estímulos son siempre sustantivos, las palabras asociadas son de selección libre, es decir, los informantes pueden relacionar a la palabra estímulo con cualquier palabra sin importar su categoría gramatical.

Los estímulos se dividieron en dos listas A y B de 117 palabras cada una. Para cada estímulo y sus asociados, los autores investigaron diferentes medidas. Entre ellas, las más relevantes para nuestro trabajo son tiempo, frecuencia y fuerza de asociación.

#### 4. Representaciones distribuidas de palabras sobre el NAP

El grafo que representa el corpus NAP se define formalmente como  $G = \{V, E, \phi\}$  donde:

- $V = \{v_i | i = 1, \dots, n\}$  es un conjunto finito de nodos de longitud  $n$ ,  $V \neq \emptyset$ , que corresponde a los estímulos y sus asociados.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leq i, j \leq n\}$ , es el conjunto de aristas.
- $\phi : E \rightarrow \mathbb{R}$ , es una función de peso sobre los ejes.

Hemos experimentado con grafos dirigidos y no dirigidos. En los grafos dirigidos, cada par de nodos  $(v_i, v_j)$  sigue un orden establecido donde el nodo inicial  $v_i$  corresponde a la palabra estímulo y el nodo final  $v_j$  a una palabra asociada. Para el grafo no dirigido, se toman todos los estímulos y se conectan con todas las palabras asociadas sin ningún orden de precedencia. Evaluamos tres funciones de peso para los ejes:

<sup>3</sup> <http://www.eat.rl.ac.uk/>

<sup>4</sup> <http://web.usf.edu/FreeAssociation>

**Tiempo** Mide los segundos que el participante tarda en dar una respuesta para cada *estímulo*.

**Frecuencia** Establece el número de ocurrencias de cada una de las palabras asociadas a un *estímulo*.

**Fuerza de asociación** Relaciona la frecuencia con el número de respuestas para cada estímulo. Se calcula de la siguiente manera: siendo  $AW$  la frecuencia de una palabra determinada asociada a un *estímulo*, y  $\Sigma F$  la suma de las frecuencias de las palabras conectadas el mismo *estímulo* (el número total de respuestas), la fuerza de asociación (AS) de la palabra  $W$  a dicho *estímulo* se obtiene con la fórmula:

$$AS_W = \frac{AW * 100}{\Sigma F}.$$

#### 4.1. Node2vec

El algoritmo *node2vec* [9] encuentra un mapeo  $f : V \rightarrow \mathbb{R}^d$  que transforma los nodos de un grafo en vectores de  $d$ -dimensiones. Define un vecindario en una red  $N_s(u) \subset V$  para cada nodo  $u \in V$  a través de una estrategia de muestreo  $S$ . El objetivo del algoritmo es maximizar la probabilidad de observar nodos subsecuentes en una camino aleatorio de una longitud fija.

La estrategia de muestreo diseñada en *node2vec* permite explorar vecindarios con caminos aleatorios sesgados. Los parámetros  $p$  y  $q$  controlan el cambio entre las búsquedas en anchura (BFS) y en profundidad (DFS) en el grafo. Así pues, elegir un equilibrio adecuado permite preservar tanto la estructura de la comunidad como la equivalencia entre nodos estructurales en el nuevo espacio vectorial.

En este trabajo, hemos usado la implementación disponible en la web<sup>5</sup> del proyecto *node2vec* con valores por defecto para todos los parámetros. Se ha examinado la calidad de los vectores con diferentes número de dimensiones  $d$ .

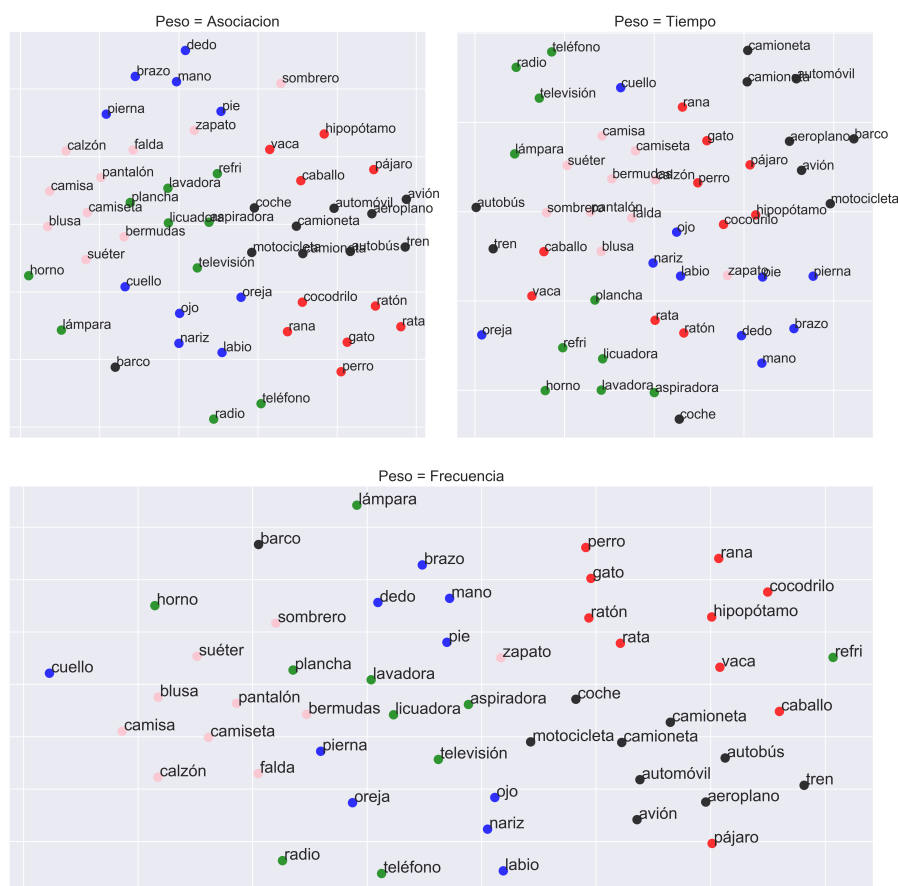
## 5. Evaluación de los vectores de palabras

Existen diversos métodos de evaluación para técnicas de vectorización de palabras no supervisadas [18], categorizadas como extrínsecas e intrínsecas. En la evaluación extrínseca, se evalúa la calidad de los vectores de palabras en tareas de procesamiento del lenguaje natural (PLN) [9, 10] y se mide la mejora en el rendimiento en la tarea evaluada. La evaluación intrínseca mide la capacidad de los vectores de palabras de capturar relaciones sintácticas o semánticas [2].

La hipótesis de la evaluación intrínseca es que palabras similares deberían tener representaciones similares. Entonces, para evaluar la similitud, primero se llevó a cabo una visualización de una muestra de palabras usando la proyección T-SNE de los vectores de palabras en un espacio vectorial bi-dimensional. En la Figura 1 se aprecia como se agrupan las palabras que están relacionadas entre

<sup>5</sup> <http://snap.stanford.edu/node2vec/>

sí. Se muestran los resultados obtenidos con las tres formas de pesado de aristas, y se observa que todas son capaces de detectar algunas coincidencias en el significado. Las figuras ilustran algunos fenómenos interesantes. Por ejemplo, cuando se toma la frecuencia como peso (la gráfica de abajo), la palabra “pájaro” se dibuja muy cerca de “avión”. De aquí se infiere que la característica “volar” es más representativa que “animal” para el modelo. Por su parte, la palabra “Caballo”, se representa más cercano a “camioneta” que a otros animales, incidiendo más en su condición de “medio de transporte”.



**Fig. 1.** Proyección de los vectores de palabras para 5 grupos semánticos (de diez palabras cada uno). Los colores están codificados como sigue: animales - rojo, transporte - negro, partes del cuerpo - azul, electrodomésticos - verde y ropa - rosa.

Además, evaluamos la capacidad de los vectores de palabras para capturar las relaciones semánticas mediante una tarea de similitud de palabras. Específicamente, usamos un subconjunto (150 pares de palabras) del corpus *WordSim*-

353 [7] compuesto por pares de términos semánticamente relacionados con puntuaciones de similitud dadas por humanos. *Hassan y Mihalcea* [10] elaboraron una versión de este corpus en español <sup>6</sup>.

Nosotros calculamos la similitud coseno entre los vectores del subconjunto de pares de palabras contenidos en el corpus *WordSim-353* y lo comparamos con la similitud dada por humanos. Las Tablas 1 y 2 presentan la correlación de Spearman, en porcentajes, de la similitud dada por etiquetadores humanos, con la similitud obtenida con vectores de palabras (aprendidos del NAP) de diferentes dimensiones aprendidos en los grafos dirigidos y no dirigidos, respectivamente.

**Tabla 1.** Correlación de Spearman (%) de la similitud coseno calculada con vectores obtenidos del grafo dirigido.

Tamaño del vector	Frecuencia	Asociación	Tiempo
300	-3.07	-3.11	-3.11
200	-1.95	-1.99	-2.03
128	0.88	0.98	0.96
100	<b>4.61</b>	<b>4.61</b>	<b>4.63</b>
50	2.51	2.42	2.39
25	-3.79	-3.89	-3.92

**Tabla 2.** Correlación de Spearman (%) de la similitud coseno calculada con vectores obtenidos del grafo no dirigido.

Tamaño del vector	Frecuencia	Asociación	Tiempo
300	43.62	43.58	50.77
200	42.89	40.55	44.67
128	39.54	44.01	50.31
100	44.66	44.31	46.50
50	45.60	<b>47.52</b>	<b>53.42</b>
25	<b>47.71</b>	45.75	51.04

Se puede observar que los vectores que se obtienen con los grafos dirigidos no son capaces de trasladar los vecindarios de los nodos al espacio vectorial. En cambio, a causa de la naturaleza no restringida de las aristas, el algoritmo *node2vec* es capaz de caminar diferentes vecindarios en el grafo no dirigido, y por ello consigue mejores representaciones vectoriales.

La tabla 3 muestra la correlación de Spearman entre la similitud coseno obtenida con los vectores pre-entrenados de *word2vec* y la similitud de los humanos (obtenida del corpus *WordSim-353*).

<sup>6</sup> <http://web.eecs.umich.edu/~mihalcea/downloads.html>

El valor de correlación más alto fue obtenido con los vectores entrenados en el *Spanish Billion Word Corpus* [5] (w2v-1b). Los vectores entrenados con la Wikipedia<sup>7</sup> en español (w2v-wk) obtuvieron resultados similares a los de nuestro método. Los mejores resultados con los vectores entrenados con *node2vec* basados en el NAP se registraron con el grafo no dirigido, considerando el *tiempo* como medida de peso de las aristas.

**Tabla 3.** Comparación con vectores pre-entrenados.

Fuente	Tamaño del vector	Correlación de Spearman
w2v-1b	300	<b>62.20</b>
w2v-wk	300	53.37
n2v-time	300	50.77
n2v-time	50	53.42

## 6. Conclusiones

Este artículo propone el uso de corpus de Normas de Asociación de Palabras en lugar de grandes corpus para obtener vectores de palabras. Para ello, se ha aplicado el algoritmo *node2vec* a un grafo construido sobre el NAP para el español de México, una pequeña colección con 4704 nodos.

Los experimentos muestran mejores resultados con grafos no-dirigidos. Se ha otorgado peso a las aristas teniendo en cuenta tres criterios diferentes: *tiempo*, *frecuencia* y *fuerza asociativa*. Los mejores resultados han sido los obtenidos con la categoría tiempo. Visto desde la perspectiva del funcionamiento del sistema *node2vec*, esto no debería ser una sorpresa. Las palabras con una índice más alto de asociación normalmente tienen un tiempo de formulación más breve, y el algoritmo busca los caminos más cortos. Como trabajo futuro, se propone repetir el experimento realizando ajustes a las variables de frecuencia y fuerza asociativa para obtener resultados más concluyentes.

Los resultados que reportamos son comparables a los obtenidos con *word2vec* entrenado con grandes corpus. El rendimiento incluso mejora los resultados alcanzados con *word2vec* entrenados en wikipedia. Sin embargo, algunas estrategias simples ayudarían a mejorar nuestros resultados. Algunas de ellas serían ajustar los parámetros del algoritmo y adaptar el sistema a diferentes tipos de vecindarios para los nodos, que podrían producir diferentes configuraciones de los vectores.

Las evaluaciones realizadas con los vectores generados con el corpus NAP mostraron resultados prometedores respecto a los índices de similitud y relacio-

<sup>7</sup> Vectores de palabras de más de 30 lenguajes:  
<https://github.com/Kyubyong/wordvectors>



alidad, queda como trabajo a futuro la evaluación de estos vectores en alguna tarea de Procesamiento de Lenguaje Natural.

**Agradecimientos.** Este trabajo ha sido realizado gracias al apoyo de los proyectos: Conacyt FC-2016-01-2225 y PAPIIT IA400117, IN403016.

## Referencias

1. Arias-Trejo, N., Barrón-Martínez, J.B., Alderete, R.H.L., Aguirre, F.A.R.: Corpus de normas de asociación de palabras para el español de México [NAP]. Universidad Nacional Autónoma de México (2015)
2. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52<sup>nd</sup> Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 238–247 (2014), <http://www.aclweb.org/anthology/P14-1023>
3. Bel-Enguix, G., Rapp, R., Zock, M.: A graph-based approach for computing free word associations. In: Proceedings of the 9<sup>th</sup> edition of the Language Resources and Evaluation Conference. pp. 221–230 (2014)
4. Borge-Holthoefer, J., Arenas, A.: Navigating word association norms to extract semantic information. In: Proceedings of the 31<sup>st</sup> Annual Conference of the Cognitive Science Society (2009)
5. Cardellino, C.: Spanish Billion Words Corpus and Embeddings (March 2016), <http://crscardellino.me/SBWCE/>
6. De Deyne, S., Navarro, D.J., Storms, G.: Associative strength and semantic activation in the mental lexicon: Evidence from continued word associations. In: Proceedings of the 35<sup>th</sup> Annual Conference of the Cognitive Science Society. Cognitive Science Society (2013)
7. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppín, E.: Placing search in context: The concept revisited. In: Proceedings of the 10<sup>th</sup> International Conference on World Wide Web. pp. 406–414. ACM (2001)
8. Garimella, A., Banea, C., Mihalcea, R.: Demographic-aware word associations. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2285–2295 (2017)
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22<sup>nd</sup> ACM International Conference on Knowledge Discovery and Data Mining. pp. 855–864. ACM (2016)
10. Hassan, S., Mihalcea, R.: Cross-lingual semantic relatedness using encyclopedic knowledge. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. pp. 1192–1201. Association for Computational Linguistics (2009)
11. Istifci, I.: Playing with words: a study of word association responses. Journal of International Social Research 0 1 (2010)
12. Jackson-Maldonado, D., Thal, D., Fenson, L., Marchman, V., Newton, T., Conboy, B.: Macarthur inventarios del desarrollo de habilidades comunicativas (inventarios): User's guide and technical manual. Baltimore, MD: Brookes (2003)
13. Kent, G.H., Rosanoff, A.J.: A study of association in insanity. Amer J. Insanity 1910(67), 317–390 (1910)

14. Kiss, G., Armstrong, C., Milroy, R., Piper, J.: An associative thesaurus of English and its computer analysis. Edinburgh University Press, Edinburgh (1973)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Computing Research Repository arXiv:1301.3781 (2013), <https://arxiv.org/abs/1301.3781>
16. Nelson, D.L., McEvoy, C.L., Schreiber, T.A.: Word association rhyme and word fragment norms. The University of South Florida (1998)
17. Sahlgren, M.: The distributional hypothesis. Italian Journal of Disability Studies 20, 33–53 (2008)
18. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 298–307 (2015)
19. Sinopalnikova, A., Smrz, P.: Word association thesaurus as a resource for extending semantic networks. In: Communications in Computing. pp. 267–273 (2004)

# Medidas de similitud semántica aplicadas a una ontología de dominio

Aimee Cecilia Hernández García, Mireya Tovar Vidal,  
José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación, Puebla,  
México

hernandez.aimee@outlook.com, mtovar@cs.buap.mx,  
jlavallentor@gmail.com

**Resumen.** La similitud semántica se utiliza para conocer si dos conceptos son semejantes en cuanto a su significado en una ontología de dominio. En esta investigación, se propone un algoritmo para evaluar las relaciones taxonómicas existentes en una ontología de Inteligencia Artificial (IA), a través de la medida de exactitud. En este caso, para evaluar las relaciones, se emplearon tres medidas de similitud semántica basadas en conocimiento: Path, Wu y Palmer y Li. Los resultados experimentales indican que de acuerdo a la medida Path la ontología tiene el 88 % de relaciones taxonómicas correctas, la medida de Wu y Palmer indica que solo el 85 % son correctas y Li indica que el 84 % son correctas. Adicionalmente, definimos una similitud promedio, a partir de estas medidas, logrando un 92 % de exactitud para este tipo de relaciones semánticas. Comparando los resultados experimentales con las respuestas de validación de un experto de dominio el sistema concuerda en un 85 %.

**Palabras clave:** similitud semántica, ontología, relaciones taxonómicas.

## Semantic Similarity Measures Applied to a Domain Ontology

**Abstract.** The semantic similarity is used to know whether two concepts are similar with respect to their meaning in a domain ontology. In this article, an algorithm to assess taxonomic relationships in an ontology of Artificial Intelligence is proposed, this is done through the accuracy measure. In order to assess the relationships, three semantic similarity measures based on knowledge are used: Path, Wu & Palmer, and Li. The experimental results show that according to the Path measure the ontology has an 88 % of right taxonomic relationships, with Wu & Palmer an 85 % is got, and the Li measure gives an 84 %. Computing the mean of these results, a 92 % of accuracy is reached for this kind of semantic

relationships. Comparing the experimental results with the assessment done by a human expert, an 85 % of agreement is found.

**Keywords:** semantic similarity, ontology, taxonomic relationships.

## 1. Introducción

En tiempos actuales, la similitud semántica ha ganado importancia en áreas como el procesamiento del lenguaje natural, la inteligencia artificial, la biomedicina, la psicología, entre otras. Debido a la enorme cantidad de datos generados electrónicamente y dado a que estos datos no se encuentran de una forma estructurada es difícil su procesamiento para obtener información útil. Por lo tanto, se han desarrollado métodos para estructurar estos datos, por ejemplo: desde bases de datos u otras representaciones como son las ontologías; para recuperar información relevante se usan métodos de procesamiento del lenguaje natural o de recuperación de información.

La similitud semántica entra en acción ante la problemática de la ambigüedad y variación lingüística en el lenguaje natural, además tiene múltiples aplicaciones como en la web semántica, en la búsqueda de respuestas, en la desambiguación del sentido de las palabras, en el reconocimiento de entidades nombradas, en la traducción automática, en la respuesta a preguntas, etc.

En este artículo se aplicarán algunas medidas de similitud semántica a una ontología de dominio. Chabot [1] define una ontología como: “Modelo de representación del conocimiento utilizado especialmente en las áreas de Web Semántica e Inteligencia Artificial. Las ontologías se usan para representar conocimiento de dominio utilizando conceptos, relaciones y axiomas”. Gruber [2] define a una ontología como “una especificación explícita y formal de una conceptualización compartida”. En general, este tipo de recurso semántico está formado por conceptos o clases, relaciones, instancias, atributos, axiomas, restricciones, reglas y eventos. Las ontologías de dominio son un sistema de representación del conocimiento que se puede organizar en estructuras taxonómicas y no taxonómicas de conceptos de algún área o dominio de conocimiento específico. Cuando una ontología contiene relaciones de tipo “is-a”, por ejemplo, una clase A es una subclase de B, se dice que tiene una relación taxonómica.

En la actualidad existen propuestas de sistemas computacionales para la generación automática de ontologías, pero, en la mayoría de los casos carecen de una evaluación automática, por lo que regularmente se desconoce la calidad de los recursos semánticos que estos sistemas generan.

La evaluación de ontologías es una tarea que consiste en medir la calidad de estos recursos. El objetivo final de la evaluación de la ontología es facilitar la labor del ingeniero del conocimiento o del experto del dominio para verificar la calidad de la misma, debido a que cuando la ontología es de un tamaño considerable, esta tarea consume mucho tiempo (horas-persona). El proceso de evaluación no suele ser trivial, pues es necesario elegir qué elementos de la ontología deberían

considerarse en el proceso de medición de la calidad de la misma, así como los criterios específicos a usar.

La similitud semántica es una medida para conocer la relación entre dos conceptos o palabras basado en sus significados, mide la distancia entre ellos, mientras menor sea la distancia más similares son los conceptos, y el resultado es expresado numéricamente. La similitud semántica se puede aplicar a una ontología dada y permite que se pueda conocer, si dos conceptos en la ontología son semánticamente similares.

El objetivo de esta investigación es implementar en un lenguaje de programación algunas medidas de similitud semántica propuestas en la literatura, y evaluar las relaciones de tipo “is-a” en una ontología del dominio de inteligencia artificial. Con la finalidad de medir el grado de relación existente entre cada par de conceptos que modelan una relación taxonómica y emitir un juicio de calidad automáticamente.

Este artículo se estructura de la siguiente manera, en la Sección 2 se presentan algunos trabajos relacionados con los métodos de similitud semántica. En la sección 3 se exponen algunas medidas de similitud propuestas en la literatura; en la Sección 4 se presenta el algoritmo propuesto, en la Sección 5 se exponen los resultados de la investigación y finalmente en la Sección 6 se presentan las conclusiones y el trabajo a futuro de esta investigación.

## **2. Trabajos relacionados**

A continuación se describen los trabajos de algunos autores que han desarrollado medidas de similitud semántica aplicadas a relaciones taxonómicas (“is-a”).

En [3] proponen una métrica llamada Distancia que evalúa el camino más corto entre dos conceptos en una base de conocimientos jerárquica, especialmente con relaciones “is-a” y así conocer la distancia entre los conceptos, esta métrica tiene sus bases en la teoría de activación propagante.

En [4] se propone una medida de similitud semántica para resolver el problema de selección léxica en la traducción automática. Los autores definen la similitud de dos conceptos que se da por la cercanía de la relación en la jerarquía. Para el cálculo de la similitud de los dos conceptos  $C_1$  y  $C_2$ , primero se calcula el número de nodos del camino de cada concepto  $C_1$  y  $C_2$  al superconcepto menos común y después se calcula el número de nodos del superconcepto menos común a la raíz.

En [5] se introduce una medida de similitud semántica que combina la propuesta del camino más corto entre un concepto a otro con el contenido de información, que es la probabilidad de ocurrencia de un concepto en un corpus. Por lo que esta medida mejora la propuesta tradicional del camino más corto, donde sólo se tiene información a nivel taxonomía y se añade el contenido de información desde el corpus a su factor de decisión.

En [6] se propone una medida de similitud en base al contenido de la información que se puede aplicar a dominios diferentes. Esta medida quiere lograr dos objetivos: el primer objetivo es la universalidad, que se refiere a que se

pueda aplicar a muchos dominios diferentes, siempre y cuando el dominio tenga un modelo probabilístico, y el segundo objetivo es la justificación teórica, que quiere decir que la medida no se define con una sola fórmula sino con un conjunto de suposiciones sobre la similitud.

En [7] se plantea una medida de similitud semántica para una taxonomía con relaciones “is-a” que se basa en el contenido de información compartida. La evaluación experimental indica que esta medida tiene mejores resultados comparados al enfoque de conteo de aristas. Definen que en un conjunto de conceptos en una taxonomía de tipo “is-a”, la clave para obtener la similitud entre dos conceptos es la información que comparten, con este enfoque se dice que mientras más abstracto es un concepto, menor es el contenido de información, entonces mientras más contenido de información compartan, más similares son.

En [8] se propone una medida de similitud semántica para una taxonomía de tipo “is-a” y “has-a” que combina el camino más corto, la profundidad del subsumidor, (subsumidor es el ancestro común más específico de dos conceptos en una ontología [7]), y la densidad semántica local, donde el camino más corto y la profundidad del subsumidor se obtiene de una base de datos léxica, la densidad semántica local se obtiene del corpus.

En [9] se define un método de similitud semántica en grafos de conocimiento, llamado *wpath*, en este trabajo se describe que hay métodos que están basados en medir la similitud semántica en base a la distancia del camino más corto entre dos conceptos en una taxonomía. Por otra parte, hay métodos que contemplan el contenido de información de los conceptos desde el corpus para mejorar el resultado de similitud semántica. Entonces lo que los autores proponen es combinar el método de medir la distancia entre conceptos y el contenido de información calculado desde el grafo de conocimiento y no desde un corpus.

Por otro lado, en Tovar et al. [10-17] se ha llevado a cabo la evaluación o validación de relaciones semánticas en ontologías, por medio de enfoques basados en patrones o por análisis formal de conceptos utilizando corpora de dominio.

En esta investigación, en base a las medidas de similitud semánticas [3-5] y el trabajo realizado en [18] se evalúa las relaciones taxonómicas “is-a” de pares de conceptos en una ontología del dominio de Inteligencia Artificial. Usando estas medidas de similitud semántica se calcula el grado de relación que existe en esos pares de conceptos que representan una relación taxonómica.

### 3. Medidas de similitud semántica

La similitud semántica se define como la estimación del parecido taxonómico de dos términos, basados en la evaluación de las evidencias semánticas comunes extraídas de una o varias fuentes de conocimiento [19] (por ejemplo, corpus textual, tesauro, taxonomías/ontologías, etc.).

En la literatura se han propuesto diferentes tipos de medidas, éstas se pueden clasificar en dos grupos: los basados en corpus y los basados en conocimiento. Las medidas basadas en corpus miden la similitud semántica entre conceptos basándose en la información obtenida de un corpus, mientras que las medidas

basadas en conocimiento miden la similitud semántica de conceptos en grafos de conocimiento [9].

En este artículo se aplicarán algunas medidas de similitud basadas en conocimiento que son las que se pueden utilizar en estructuras de representación del conocimiento como las ontologías. Basándonos en el trabajo realizado en [9] el objetivo es implementar algunas de las medidas de similitud haciendo una adaptación en el lenguaje de programación Python, usando las bibliotecas NLTK [20], RDFlib<sup>1</sup> y el framework Sematch [18], para una ontología en particular, se obtendrán los resultados de los cálculos de cada medida y se evaluarán los resultados obtenidos.

A continuación se describen las medidas de similitud que se implementaron para esta investigación.

### 3.1. Path

En [3] se propone una medida llamada Distancia, y en base a esta medida en el trabajo de [9] se definió la ecuación  $sim_{Path}$  (ver Ecuación (1)), usando el camino más corto ( $length$ ) entre dos conceptos  $c_i$  y  $c_j$ , mediante esta distancia se puede saber la similitud entre los conceptos en una taxonomía:

$$sim_{path}(c_i, c_j) = \frac{1}{1 + length(c_i, c_j)}. \quad (1)$$

Para las siguientes medidas presentadas en el trabajo de [9] se necesitan dos conceptos para implementarlas, primeramente se define el concepto de profundidad o  $depth$ . La profundidad de un concepto ( $depth(c_i)$ ) es el camino más corto desde un concepto  $c_i$  al concepto raíz  $c_{raiz}$  y se define en la Ecuación (2):

$$depth(c_i) = length(c_i, c_{raiz}). \quad (2)$$

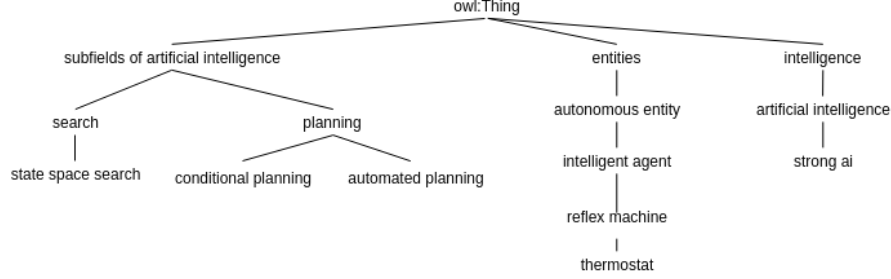
El segundo concepto es el subsumidor menos común o *Least Common Subsumer (LCS)* que en [9] se define como el concepto más específico que es ancestro común de dos conceptos. Por ejemplo, en la Fig. 1 el LCS de los conceptos *conditional planning* y *state space search* es *subfields of artificial intelligence*.

Por ejemplo, si  $c_i = conditional\ planning$  y  $c_j = state\ space\ search$ , el resultado de esta medida de similitud es 0.20.

### 3.2. Wu y Palmer

En base a las definiciones antes presentadas y a la investigación de [4] se define la medida de Wu y Palmer en [9], donde se mide la similitud de dos conceptos  $c_i$  y  $c_j$  donde se mide la distancia más corta de cada concepto  $c_i$  y  $c_j$  con el concepto raíz y la distancia del LCS de cada concepto  $c_i$  y  $c_j$  con el concepto raíz.

<sup>1</sup> <https://github.com/RDFLib/rdfliib>



**Fig. 1.** Fragmento de la taxonomía de IA [21].

$$sim_{wup}(c_i, c_j) = \frac{2 * depth(c_{lcs})}{depth(c_i) + depth(c_j)}, \quad (3)$$

donde  $depth(c_i)$  y  $depth(c_j)$  es la distancia más corta de cada concepto con el concepto raíz y  $depth(c_{lcs})$  es la distancia del LCS con el concepto raíz.

Como ejemplo, para los conceptos  $c_i = conditional\ planning$  y  $c_j = state\ space\ search$  (ver Fig. 1), el valor de  $depth(c_{lcs})$  es de 2, mientras que los valores para  $depth(c_i)$  y  $depth(c_j)$  es de 4 para cada uno, si  $depth(c_{raiz}) = 1$ , entonces el cálculo de la similitud semántica utilizando la Ecuación (3) es de 0.5, ver resultado en la Ecuación (4):

$$sim_{wup}(c_i, c_j) = \frac{2 * 2}{4 + 4} = 0,5. \quad (4)$$

### 3.3. Li

En base al método diseñado por Li [8] se formula la Ecuación 5 propuesta por [9], donde se combina el camino más corto ( $depth$ ) de ambos conceptos  $c_i$  y  $c_j$  y el LCS ( $c_{lcs}$ ) de los conceptos, para calcular su similitud:

$$sim_{li}(c_i, c_j) = e^{-\alpha length(c_i, c_j)} \cdot \frac{e^{\beta depth(c_{lcs})} - e^{-\beta depth(c_{lcs})}}{e^{\beta depth(c_{lcs})} + e^{-\beta depth(c_{lcs})}}, \quad (5)$$

donde  $\alpha$  es un parámetro que contribuye a la longitud del camino y  $\beta$  es el parámetro para la profundidad del camino. De acuerdo con el trabajo de [8], el parámetro óptimo para  $\alpha$  es 0.2 y para  $\beta$  es 0.6.

Por ejemplo, para los conceptos  $c_i = conditional\ planning$  y  $c_j = state\ space\ search$  (ver Fig. 1), el valor para  $length(c_i, c_j)$  es de 4 y el valor para  $depth(c_{lcs})$  es de 2, con estos valores se calcula la similitud semántica de este par de conceptos y se obtiene el resultado de 0.374 (ver Ecuación 6):

$$sim_{li}(c_i, c_j) = e^{-0,2*4} \cdot \frac{e^{0,6*2} - e^{-0,6*2}}{e^{0,6*2} + e^{-0,6*2}} = 0,3745. \quad (6)$$



Dados los resultados obtenidos con el ejemplo y las medidas de similitud, observamos que los conceptos no son tan similares y eso lo podemos confirmar al observar que se encuentran en diferentes ramas de la taxonomía de la Figura 1, aún teniendo un ancestro común, es decir, entre los dos conceptos no existe una relación directa de tipo “is-a”.

#### 4. Algoritmo propuesto

En esta sección se presenta un algoritmo para la evaluación de relaciones taxonómicas de una ontología de dominio utilizando la medida de exactitud, la cual se presenta en la Ecuación 7. Donde el *Total de casos* es el total de relaciones taxonómicas existentes en la ontología de dominio y el *Total de casos correctos* son las relaciones consideradas por el algoritmo como relaciones taxonómicas:

$$Exactitud = \frac{Cantidad\ de\ casos\ correctos}{Total\ de\ casos}. \quad (7)$$

El funcionamiento general del Algoritmo 1 consiste en: Por cada par de conceptos se calculan las tres medidas de similitud semántica. Después, se calcula un umbral por cada medida, llamado ( $umbral_{medida}$ ), y es la suma de los resultados de similitud por cada par de conceptos dividido entre el número de relaciones taxonómicas. Si el resultado de la medida de similitud para ese par de conceptos supera el  $umbral_{medida}$ , entonces la relación taxonómica es verdadera, de lo contrario es falsa. Posteriormente, aplicamos la medida de exactitud al total de relaciones taxonómicas. Por otro lado, calculamos un promedio de umbrales ( $umbral_{promedio}$ ) y se realiza el mismo procedimiento, es decir, si el promedio de las similitudes para ese par de conceptos supera el  $umbral_{promedio}$ , entonces la relación taxonómica es verdadera, de lo contrario es falsa. Nuevamente, calculamos la exactitud para estos resultados. A continuación se describe con mayor detalle los pasos del algoritmo.

En el Algoritmo 1 se utiliza el llamado al *framework Sematch* propuesto en [18] implementado en el lenguaje de programación Python y publicado en GitHub<sup>2</sup>. Los datos de entrada en este algoritmo son:  $|RT|$  que se refiere al total de relaciones “is-a”, la lista de conceptos y subconceptos de la ontología de entrada, la lista de validación de las relaciones por parte de un experto de dominio. La salida es la evaluación de las relaciones “is-a”, por cada medida de similitud utilizada: Path, Wu y Palmer y Li.

Del paso 1-4 se realiza el llamado al *framework Sematch*, donde  $sim_{medida}$  corresponde a una de las tres medidas utilizadas, los resultados se almacenan en una lista  $res_{medida}$ . Antes de estos pasos, el *framework* requiere la ontología de Inteligencia Artificial en formato OWL (ai.owl). Esto no es parte del algoritmo, es un requerimiento del *framework* para el funcionamiento de las medidas de similitud semántica.

Continuando con el Algoritmo 1, en el paso 5 se calcula el umbral para cada medida de similitud semántica.

<sup>2</sup> <https://github.com/gsi-upm/sematch/>

---

**Algoritmo 1** Algoritmo propuesto

---

**Entrada:** concepto[], subConcepto[], experto[], |RT|, ai.owl

**Salida:** Exactitud de cada medida de similitud semántica

```

1: {Llamar al framework Sematch para usar las funciones de similitud semántica.}
2: para i=0 hasta |RT| hacer
3:    $res_{medida}[i] \leftarrow sim_{medida}(concepto[i], subConcepto[i])$ 
4: fin para
5:  $umbral_{medida} \leftarrow promedio(res_{medida}[])$ 
6:  $umbral_{promedio} \leftarrow promedio(umbral_{medida})$ 
7:  $tablaVerdad_{medida}[] \leftarrow compUmbral(res_{medida}[], umbral_{medida}, |RT|)$ 
8:  $exactSistema_{medida}[] \leftarrow promedio(tablaVerdad_{medida}[])$ 
9:  $exactitud_{medida} \leftarrow exactMedida(tablaVerdad_{medida}[], experto[], |RT|)$ 
10:  $tablaPromGeneral[] \leftarrow promedioSist(res_{path}, res_{wup}, res_{li}, umbral_{promedio}, |RT|)$ 
11:  $PromGeneral \leftarrow promedio(tablaPromGeneral[])$ 
12:  $PromExperto \leftarrow exactMedida(tablaPromGeneral[], experto[], |RT|)$ 

```

---

Para cada lista obtenida en el paso 2, se suman los resultados almacenados y se divide entre el número de relaciones, es decir, se calcula el promedio. En el paso 6, se calcula el  $umbral_{promedio}$  que es el resultado del promedio de los tres umbrales.

En el paso 7 se llama a la función *compUmbral*, que compara los resultados de similitud de cada medida contra el umbral de esa medida, que se detalla en el Algoritmo 2. Esta función toma como entrada la lista de resultados ( $res_{medida}$ ) y el umbral por cada medida, así como el número de relaciones “is-a” en la ontología y el algoritmo regresa la lista  $tablaVerdad_{medida}$ . Esta lista es el resultado de comparar el valor de similitud del par de conceptos contra el umbral, si el valor de similitud es mayor que el umbral, el valor de verdad que toma esa relación es verdadero (1 en el algoritmo) de lo contrario es falso (0 en el algoritmo).

---

**Algoritmo 2** Función *compUmbral*

---

**Entrada:**  $res_{medida}[], umbral_{medida}, |RT|$

**Salida:**  $tablaVerdad_{medida}[]$

```

1: para i=0 hasta |RT| hacer
2:   si  $res_{medida}[i] \geq umbral_{medida}$  entonces
3:      $tablaVerdad_{medida}[i] \leftarrow 1$ 
4:   si no
5:      $tablaVerdad_{medida}[i] \leftarrow 0$ 
6:   fin si
7: fin para
8: devolver  $tablaVerdad_{medida}[]$ 

```

---

En el paso 8, se obtiene la exactitud que tiene el sistema con cada medida de similitud, en este paso se calcula el promedio de los resultados de la lista

*tablaVerdad*, por cada medida. En el paso 9, se procede a calcular la exactitud de cada medida y se realiza llamando a la función *exactMedida* que se detalla en el Algoritmo 3. Esta función toma como entrada la *tablaVerdad* de cada medida, la lista anotada de esas relaciones por un Experto en el tema y el número total de relaciones. Si el Experto y el valor almacenado en *tablaVerdad<sub>medida</sub>* coinciden, se asigna el valor 1 (verdadero) a la lista *exactitudTabla<sub>medida</sub>*, de otro modo se asigna un 0 (falso). Al terminar, la función regresa la exactitud de los resultados obtenidos de la lista *exactitudTabla<sub>medida</sub>*, obteniendo como resultado la Exactitud de cada medida de similitud (*exactitud<sub>medida</sub>*).

---

**Algoritmo 3** Función *exactMedida*


---

**Entrada:** *tablaVerdad<sub>medida</sub>*[], *experto*[],  $|RT|$

**Salida:** *exactitud<sub>medida</sub>*

```

1: para i=0 hasta  $|RT|$  hacer
2:   si tablaVerdadmedida[i] = 1 y experto[i] >= 1 entonces
3:     exactitudTablamedida[i]  $\leftarrow$  1
4:   si no, si tablaVerdadmedida[i] = 0 y experto[i] = 0 entonces
5:     exactitudTablamedida[i]  $\leftarrow$  1
6:   si no
7:     exactitudTablamedida[i]  $\leftarrow$  0
8:   fin si
9: fin para
10: exactitudmedida  $\leftarrow$  Exactitud(exactitudTablamedida[],  $|RT|$ )
11: devolver exactitudmedida

```

---

En el paso 10, se hace el llamado a la función *promedioSist* que se muestra en el Algoritmo 4, esta función toma como entrada las listas de los resultados de similitud de cada medida, el *umbral<sub>promedio</sub>* y el número de relaciones, después devuelve *tablaPromGeneral* con los resultados obtenidos. En el paso 11 se calcula el promedio de los resultados obtenidos en *tablaPromGeneral* para obtener el promedio general. Por último en el paso 12, se llama a la función *exactMedida* para comparar los resultados de *tablaPromGeneral* contra la lista anotada de las relaciones por el Experto.

## 5. Resultados experimentales

En esta investigación se utilizó la ontología de IA propuesta en [21]. La Tabla 1 muestra el total de conceptos ( $|C|$ ) y relaciones taxonómicas ( $|RT|$ ) existentes en la ontología de dominio.

En la Tabla 2 se muestran los resultados experimentales de las medidas de similitud aplicadas a un subconjunto de relaciones taxonómicas de la ontología de dominio. Por ejemplo, en los conceptos *RDF* y *Standard* que tienen una relación de tipo “is-a”, la medida Path indica que son 50 % similares, Wup que son un 40 % similares y Li que son un 43 % similares.

---

**Algoritmo 4** Función *promedioSist*


---

**Entrada:**  $res_{medida}[], res_{medida}[], res_{medida}[], umbral_{promedio}, |RT|$ 
**Salida:**  $tablaPromGeneral[]$ 

```

1: para i=0 hasta  $|RT|$  hacer
2:   si promedio( $res_{path}[i], res_{wup}[i], res_{li}[i]$ )  $\geq umbral_{promedio}$  entonces
3:      $tablaPromGeneral[i] \leftarrow 1$ 
4:   si no
5:      $tablaPromGeneral[i] \leftarrow 0$ 
6:   fin si
7: fin para
8: devolver  $tablaPromGeneral[]$ 

```

---

**Tabla 1.** Total de conceptos y relaciones taxonómicas de la ontología IA.

Ontología	$ C $	$ RT $
IA	233	205

Por otro lado, las tres medidas indican que los conceptos *Natural Language* y *Language* tienen una relación semántica entre ellos.

**Tabla 2.** Muestra de los resultados obtenidos por cada medida de similitud semántica.

$Concepto_1$	$Concepto_2$	$Path$	$Wup$	$Li$
Standard	RDF	0.5	0.4	0.439
Language	Natural Language	0.5	0.857	0.775
Ability	Human Cognitive Ability	0.5	0.8	0.683
Artificial Intelligence	Strong AI	0.5	0.333	0.439
Set of Inference	Representation	0.5	0.667	0.682

Sin embargo, con la finalidad de emitir un grado de similitud entre los pares de conceptos, se procedió a calcular un umbral por medida de similitud. En este caso, el umbral es el promedio de todos los resultados obtenidos de las relaciones taxonómicas de cada medida de similitud. Para tener un mejor criterio, se calculó un nuevo umbral obtenido del promedio de los umbrales de cada medida. Los umbrales obtenidos en los experimentos se muestran en la tabla 3.

Por último, en el Algoritmo 1 se calcula la exactitud de las relaciones taxonómicas, considerando cada medida de similitud semántica. El algoritmo asigna a cada par de conceptos el valor verdadero si el resultado de la medida de similitud supera el umbral, de lo contrario asigna el valor falso. Los resultados de la medida de exactitud se muestran en la Tabla 4 comparadas con los resultados de la exactitud que un experto le asignó al total de relaciones taxonómicas de la ontología de dominio. Como puede apreciarse los resultados experimentales indican que más del 84 % de las relaciones taxonómicas mantienen una relación semántica entre sí.

**Tabla 3.** Umbral para cada medida de similitud.

Path	Wup	Li	Promedio
0.5	0.779	0.68	0.653

Además, en la Tabla 4 se presenta los resultados obtenidos con el umbral promedio (Prom Sistema) que indica que las relaciones taxonomicas son un 92 % similares y que los resultados con respecto al experto este se encuentra muy cercano al mismo. Por lo tanto, consideramos que los resultados nos indican que las relaciones taxonómicas son correctas en un 92 % de exactitud.

**Tabla 4.** Exactitud obtenida para la ontología IA por cada medida de similitud.

Ontología	Experto	Path	Wup	Li	Prom Sistema	Prom Experto
IA	0.888	0.888	0.854	0.849	0.927	0.854

## 6. Conclusiones

En esta investigación se implementó un algoritmo en Python utilizando el *framework Sematch*, para la evaluación de relaciones taxonómicas de una ontología de Inteligencia Artificial, a través de tres medidas de similitud semántica basadas en conocimiento: Path, Wu y Palmer y Li. Estas medidas se basan en la distancia que existe entre un par de conceptos colocados en el grafo de la ontología. En particular se aplicaron a las relaciones de tipo “is-a” o taxonómicas. En base a los resultados experimentales, observamos que las tres medidas muestran que por lo menos el 84 % de las relaciones taxonómicas mantienen este tipo de relación semántica en la ontología.

Como trabajo a futuro se propone implementar otras medidas de similitud semántica basadas en contenido de la información. Asimismo aplicarlas a otras ontologías y compararlas con los resultados de otros expertos de dominio y realizar pruebas estadísticas, con la finalidad de emitir un juicio en cuanto a la evaluación de las relaciones semánticas y conceptos definidos en ontologías de dominio.

**Agradecimientos.** Esta investigación es apoyada por el Fondo Sectorial de Investigación para la Educación, con el proyecto CONACyT CB/257357 y por el proyecto VIEP-BUAP ID 00356, México.

## Referencias

1. Chabot, Y., Nicolle, C.: Semantic Measures: A State of the Art. In Khosrow-Pour, M., ed.: The Encyclopedia of Information Science and Technology. IGI Global, pp. 4690–4698 (2014)
2. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.* 43(5-6), pp. 907–928 (December 1995)
3. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Trans. Systems, Man, and Cybernetics* 19, pp. 17–30 (1989)
4. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. (ACL'94), pp. 133–138, Stroudsburg, PA, USA, Association for Computational Linguistics (1994)
5. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR* (1997)
6. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning. (ICML '98), pp. 296–304, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
7. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Int. Res.* 11(1), pp. 95–130 (July 1999)
8. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. on Knowl. and Data Eng.* 15(4), pp. 871–882 (July 2003)
9. Zhu, G., Iglesias, C.A.: Computing semantic similarity of concepts in knowledge graphs. *IEEE Trans. on Knowl. and Data Eng.* 29(1), pp. 72–85 (January 2017)
10. Lasserre, H.R., Tovar, M.: Proposal for automatic extraction of taxonomic relations in domain corpus. *Research in Computing Science* 133, pp. 29–39 (2017)
11. Tovar, M.: Evaluación Automática de Ontologías de Dominio Restringido. PhD thesis, Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Ciencias Computacionales (2015)
12. Tovar, M., Pinto, D., Montes, A., Serna, J.G.: An approach based in lsa for evaluation of ontological relations on domain corpora. In Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera-López, J.A., eds.: Pattern Recognition (MCPR'17), LNCS, 10267, pp. 225–233, Cham, Springer International Publishing (2017)
13. Tovar, M., Pinto, D., Montes, A., Serna, J.G.: A metric for the evaluation of restricted domain ontologies. *Computación y Sistemas* 22(1), pp. 147–162 (2018)
14. Tovar, M., Pinto, D., Montes, A., Serna, J.G., Vilariño, D.: Evaluation of ontological relations in corpora of restricted domain. *Computación y Sistemas* 19(1), pp. 135–149 (2015)
15. Tovar, M., Pinto, D., Montes, A., Serna, J.G., Vilariño, D.: Identification of ontological relations in domain corpus using formal concept analysis. *Engineering Letters* 23(2), pp. 72–76 (2015)
16. Tovar, M., Pinto, D., Montes, A., Serna, J.G., Vilariño, D.: Patterns used to identify relations in corpus using formal concept analysis. In Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Sossa-Azuela, J.H., Olvera-López, J.A., Famili, F., eds.: Pattern Recognition. (MCPR'15), LNCS, 9116, pp. 236–245, Cham, Springer International Publishing (2015)

17. Tovar, M., Pinto, D., Montes, A., Serna, J.G., Vilariño, D., Beltrán, B.: Use of lexico-syntactic patterns for the evaluation of taxonomic relations. In Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Olvera-Lopez, J.A., Salas-Rodríguez, J., Suen, C.Y., eds.: Pattern Recognition. (MCPR'14), 8495, pp. 331–340, Cham, Springer International Publishing (2014)
18. Zhu, G., Fernandez, C.A.I.: Sematch: semantic entity search from knowledge graph. In: Joint Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies and the 3rd International Workshop on Human Semantic Web Interfaces (SumPre 2015, HSWI 2015) co-located with the 12th Extended Semantic Web Conferen, 1556, pp. 1–12 (2015)
19. Batet, M., Sánchez, D.: Review on semantic similarity. In Khosrow-Pour, M., ed.: Encyclopedia of Information Science and Technology. 3rd edn. IGI Global, pp. 7575–7583 (2014)
20. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media, Inc. (2009)
21. Zouaq, A., Gasevic, D., Hatala, M.: Linguistic patterns for information extraction in ontocmaps. In: Proceedings of the 3rd International Conference on Ontology Patterns, 929, pp. 61–72, CEUR-WS. org (2012)





# Detección automática de engaño en notas de opinión a partir de técnicas de perfilado de autores

Jonathan Serrano-Pérez<sup>1</sup>, Javier Sánchez-Junquera<sup>1</sup>,  
Hugo Jair Escalante-Balderas<sup>1</sup>, Luis Villaseñor-Pineda<sup>1,2</sup>

<sup>1</sup> Instituto Nacional de Astrofísica, Óptica y Electrónica,  
Laboratorio de Tecnologías del Lenguaje, Puebla,  
México

<sup>2</sup> Université d'Artois,  
Centre de Recherche en Linguistique Française, Arras,  
France

{js.perez, hugojair, villasen}@inaoep.mx, jjsjunquera@gmail.com

**Resumen.** El presente trabajo muestra los resultados alcanzados al aplicar un método originalmente propuesto para perfilado de autores a la detección automática de engaño. El método representa cada perfil (i.e. autores de cierto género o rango de edad) a través de *subperfiles* para cada categoría. Es decir, no supone que todos los jóvenes, o todos los adultos, escriben con el mismo estilo. Este trabajo, retoma la misma suposición, e intenta afinar la discriminación entre notas engañosas y verdaderas, al suponer que existe más de un posible estilo para redactar este tipo de textos. El presente trabajo analiza el comportamiento del método variando el número de subperfiles en dos tipos de colecciones usadas para la detección del engaño: reseñas sobre hoteles y temas controversiales. El método alcanzó resultados alentadores, difiriendo los resultados según el tipo de documentos donde se pretende detectar el engaño.

**Palabras clave:** minería de textos, clasificación no-temática de textos, detección de engaño.

## Automatic Deception Detection in Opinion Notes using Author Profiling Techniques

**Abstract.** The present work shows the results achieved by applying a method originally proposed for author profiling to the automatic detection of deception. The method represents each profile (i.e. authors of a certain genre or age range) through *sub-profiles* for each category. That is, it does not assume that all young people, or all adults, write with the same style. This work takes up the same assumption, and attempts to refine the discrimination between deceptive and true notes, assuming

that there is more than one possible style to write such texts. This paper analyzes the behavior of the method by varying the number of sub-profiles in two types of collections used for the detection of deception: hotel reviews and controversial issues. The method achieved encouraging results, with results differing according to the type of documents where the deceit is intended to be detected.

**Keywords:** text mining, non-thematic text classification, deception detection.

## 1. Introducción

El uso de la internet se ha generalizado de tal forma que se recurre a este medio de información casi para cualquier cosa. En especial, consultar la internet para informarse sobre valoraciones de productos o servicios es habitual. De este modo, una persona que desea adquirir un producto o un servicio recurre a la web para responder preguntas como ¿el producto o servicio cumple con lo que promete?, ¿la tienda o el vendedor es confiable?, ¿la tienda o página web me ofrece alguna garantía por defectos o si no llega el producto?, entre otras. Básicamente, estas preguntas se responden con los comentarios o reseñas de compradores previos.

Algunos vendedores de productos y servicios, se han dado cuenta de esta situación y a través de estrategias poco éticas han intentado sacar provecho de este comportamiento al agregar comentarios positivos referentes a sus productos, y/o escribir reseñas negativas a productos o servicios de sus competidores.

La búsqueda de métodos automáticos para detectar opiniones que fueron escritas con la intención de engañar se le conoce como *detección de engaño*. La importancia de detectar automáticamente el engaño (u opiniones falsas<sup>3</sup>) es clara en situaciones como el caso de *TripAdvisor*. Dicho sitio cuenta con millones de opiniones de viajeros acerca de alojamientos, y se tiene particular interés en la detección de opiniones falsas, cuyo fin generalmente es aumentar o disminuir la reputación de un establecimiento por parte de propietarios o competidores, respectivamente<sup>4</sup>. Sin embargo, existen muchas otras situaciones en que este tipo de métodos podrían ser de ayuda al experto para la toma de decisiones, como es el caso de evaluación de veracidad de testimonios.

La detección automática del engaño recae principalmente en observar elementos que brinden evidencia de haber experimentado en *carne propia* los hechos relatados. Elementos como el uso de la primera persona, expresiones incluyendo valoraciones sensoriales, y la descripción puntual y detallada de la experiencia pueden proporcionar evidencia de la veracidad de la opinión. Trabajos previos han demostrado que estos elementos pueden capturarse a través de técnicas de

<sup>3</sup> Si bien las opiniones falsas no implican necesariamente la existencia de engaño, o sea, la intención de engañar, en este documento se mencionará “opiniones falsas” como expresión alternativa para referirse a “opiniones engañosas”.

<sup>4</sup> [https://www.tripadvisor.es/vpages/review\\_mod\\_fraud\\_detect.html](https://www.tripadvisor.es/vpages/review_mod_fraud_detect.html)

minería de texto, al capturar rasgos preponderantemente asociados al estilo de escritura de la reseña [10].

El presente trabajo explora y evalúa la aplicación de una técnica usada con éxito para la discriminación entre perfiles de autor identificando características como género o rango de edad. Dicha técnica [6] recurre al supuesto que el estilo y los temas tratados en el documento permiten discriminar, por ejemplo, a qué género pertenece el autor. Lo que es más, abre la posibilidad de capturar diversas actitudes entre autores pertenecientes a la misma clase, al considerar la existencia de *subperfiles*, ya que es difícil de imaginar que todos los autores recurren al mismo estilo de escritura. Extendiendo esta idea a la tarea de detección del engaño, no sólo deseamos discriminar entre opiniones verdaderas y falsas, sino incluso se puede suponer que el estilo en cada clase no es homogéneo. Es decir, es de suponer que existen diferentes estilos entre los autores de notas falsas así como diferentes estilos entre autores de notas verdaderas.

A continuación, en la Sec. 2, se presentan algunos trabajos relacionados a la detección de engaño. En la Sec. 3, se describe el método llevado a cabo. En la Sec. 4 se detallan los datos sobre los corpus de prueba y se muestran los resultados obtenidos junto con una breve discusión de los mismos. Finalmente se dan conclusiones preliminares en la Sec. 5.

## 2. Trabajo relacionado

Existen diferentes trabajos donde se intenta detectar las opiniones engañosas de las verdaderas. Estos trabajos difieren en las representaciones usadas así como en el tipo de documentos donde se desea hacer la detección. Respecto al tipo de documentos se identifican dos grandes tipos de colecciones: (i) opiniones *spam*<sup>5</sup> sobre productos o servicios, tales como libros, restaurantes, hoteles y doctores [10, 3, 9] y (ii) engaño en opiniones sobre tópicos controversiales como aborto, pena de muerte, y sentimientos sobre mejores amigos [8, 7, 11]. Las colecciones varían considerablemente no sólo por el tipo de contenido sino también desde el punto de vista psicológico. En la primera colección se recurrió a voluntarios para realizar el trabajo de redacción, y ellos estuvieron conscientes de que sus notas falsas no tendrían ninguna implicación. En el caso del otro tipo de colección, el autor estaba consciente de que plasmaba creencias propias y tendrían una repercusión sobre su imagen ante terceros, posibilitando así la presencia de emociones negativas vinculadas anteriormente con el acto de mentir [2, 16].

Respecto al tipo de atributos utilizados para enfrentar esta tarea, se han experimentado diferentes rasgos que se diferencian en cuanto a su complejidad y a lo que son capaces de capturar:  $n$ -gramas de palabras y de caracteres [15], estructuras sintácticas [3, 13], lista de criterios psicolingüísticos [5] y atributos semánticos [1].

<sup>5</sup> Opiniones *spam* o *fake reviews*, son opiniones engañosas, escritas de forma que parezcan auténticas, y en las que deliberadamente se da información falsa influyendo en la decisión de usuarios y clientes [2,4].

Sorprendentemente, las secuencias de  $n$  palabras (o  $n$ -gramas de palabras), han servido para discriminar engaño de no engaño con un desempeño superior al del ser humano. En [10] abordaron la tarea como una clasificación de textos mediante un clasificador basado en  $n$ -gramas de palabras. Con el propósito de modelar el contenido y el contexto, consideraron tres conjuntos de atributos: *unigramas*, la combinación de *unigramas* y *bigramas* (*bigramas*<sup>+</sup>), y la combinación de *unigramas*, *bigramas* y *trigramas* (*trigramas*<sup>+</sup>). Los autores contrastaron los resultados de la clasificación de 800 opiniones positivas sobre hoteles mediante  $n$ -gramas, frente a otro enfoque usando criterios psicolingüísticos (*LIWC*). Los resultados sugieren que con *unigramas* se discrimina mejor que con un conjunto de criterios preestablecidos como *LIWC*, y que aproximaciones sensibles al contexto (*bigramas*<sup>+</sup>) pueden mejorar la clasificación (89 % de exactitud). En [9] se mostró igualmente una mayor efectividad de *bigramas*<sup>+</sup> (86 % de exactitud) frente a las predicciones de jueces humanos, esta vez incluyendo 800 opiniones negativas.

En el caso de opiniones controversiales, en [7] se recolectaron opiniones en tópicos de pena de muerte, aborto, y sentimientos hacia el mejor amigo. En este trabajo, se aplicó un proceso de *stemming*, eliminándose las diferentes variaciones de una misma palabra y tomándolas como sinónimos. El desempeño promedio de los tres dominios fue de un 70 % de exactitud. Como se mencionó en párrafos anteriores, se trata de colecciones de características muy distintas a las opiniones *spam*, de ahí la diferencia de resultados.

Finalmente, una representación más compleja que los simples patrones léxicos fue la empleada por [3] tratando de describir más ampliamente el estilo de los engañadores. Los autores aplicaron su método en opiniones de productos, servicios y opiniones controversiales. Este consistió en el uso de reglas de producción basadas en árboles de derivación de acuerdo a gramáticas libres de contexto (*CFG*, por siglas en inglés). Con esta información fue posible detectar engaño, alcanzando aún mejores resultados cuando estos atributos se combinan con atributos léxicos (90 % de exactitud). Claro está que este método depende de recursos lingüísticos incrementando su costo computacional y restringiendo su ámbito de utilidad.

Cabe mencionar un último trabajo que antecede y motiva el presente estudio. En [13] se probaron varias representaciones entre las que destacó el discriminar primeramente por género del autor para posteriormente detectar el engaño. Los autores indican en sus experimentos que los *unigramas* fueron la representación más robusta, que las mentiras en general son más difíciles de detectar que las verdades, y que las mentiras dichas por mujeres son más fáciles de identificar que las de los hombres.

El presente trabajo no distingue entre los géneros de los autores, información que no está presente en las colecciones de prueba. Pero sí se busca discriminar el engaño de la verdad afinando la granularidad de la representación al suponer que posiblemente existen *subperfiles* tanto entre los engañadores como entre aquellos que dicen la verdad. La siguiente sección describe la representación y metodología utilizadas.

### 3. Metodología

#### 3.1. Representación de documentos

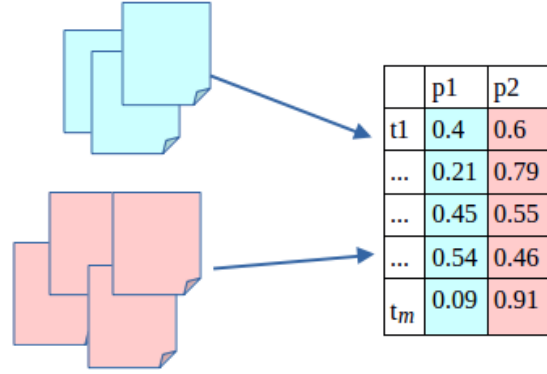
La representación usada parte directamente de lo expuesto en [6]. Esta técnica, también conocida como *SOA2*, hace la suposición que los autores de una clase de documentos están repartidos en diferentes *subperfiles*. Es por esto que se tiene interés en aplicar esta técnica en la detección del engaño, al suponer que podrían encontrarse diferentes *subperfiles* de mentirosos, lo cual podría aportar información significativa en el análisis y clasificación de nuevos documentos. Los siguientes párrafos detallan la aplicación del método *SOA2*.

**Notación.** Tomada de [6]:

- $D = \{(d_1, y_1), \dots, (d_n, y_n)\}$ ,  $D$  es una colección de  $n$  parejas de documentos ( $d_i$ ) y variables ( $y_i$ ), donde la variable representa el perfil al cual está asociado el documento.
- $y_i \in P = \{p_1, \dots, p_q\}$ ,  $P$  es el conjunto de diferentes perfiles y  $q$  es la cardinalidad de  $P$ .
- $V = \{v_1, \dots, v_m\}$ ,  $V$  es el vocabulario de la colección de documentos  $D$ .
- $v_i$  es representado como un vector  $\mathbf{t}_i \in \mathbb{R}^q$ , por lo tanto  $\mathbf{t}_i = (t_{i,1}, \dots, t_{i,q})$ , donde cada elemento  $t_{i,j}$  indica el grado de asociación entre el término  $v_i$  y el perfil  $p_j$ .
- $tf(d_k, v_i)$  es la frecuencia del término  $v_i$  en el documento  $d_k$ ,  $len(d_k)$  es la cantidad de términos que contiene el documento  $d_k$ .
- $\mathbf{x}_k$  representa al  $k$ -ésimo documento, donde  $\mathbf{x}_k \in \mathbb{R}^q$ .

**Representación de función de los perfiles.** Dado que se tiene una colección de documentos etiquetados, las diferentes etiquetas son vistas como los perfiles, por ejemplo, en el corpus de *OpSpam* se tienen las clases de engaño y veraz, por lo tanto se tendrían dos perfiles. Sabiendo esto, se llevará a cabo la representación del vocabulario del corpus, por lo que se crea una matriz del tamaño del vocabulario por la cantidad de perfiles (véase Figura 1). Como se puede observar en la Figura 1, cada perfil aporta información a las palabras que son usadas dentro del mismo, este aporte está dado por la ecuación 1. Posteriormente se hace un normalizado por perfil, es decir, se normalizan las columnas de la matriz, y finalmente se hace una normalización por término, es decir, por fila. De esta forma se construye la representación de los términos en el espacio de perfiles:

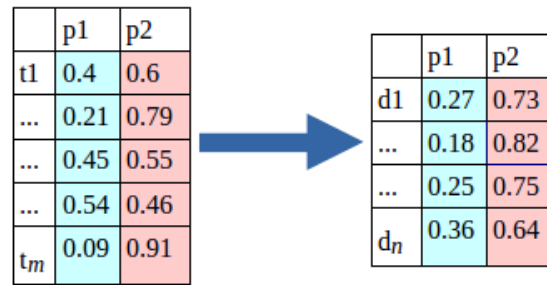
$$t_{i,j} = \sum_{\forall d_k: y_k = p_j} \log_2(1 + \frac{tf(d_k, v_i)}{len(d_k)}). \quad (1)$$



**Fig. 1.** Se crea una matriz de tamaño  $m \times q$ , donde  $m$  es el tamaño del vocabulario y  $q$  es el numero de perfiles, posteriormente los documentos de cada perfil aportan información a las palabras que son usadas dentro del mismo, este aporte esta dado por la ecuación 1. De esta forma se obtiene la representación de los términos en el espacio de perfiles.

**Representación de documentos.** Una vez que se tiene la representación de los términos en el espacio de perfiles, se usan estos para representar los documentos en el mismo espacio. Esto es, cada palabra del documento aporta información para la representación en el espacio de perfiles (véase Figura 2), este aporte está dado por la ecuación 2. Consecuentemente se obtienen las representaciones de los documentos en el espacio de perfiles:

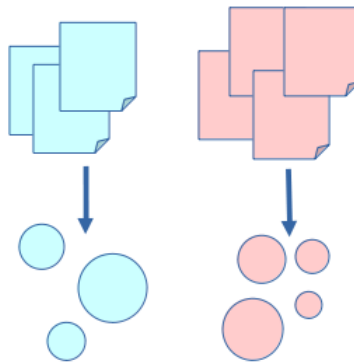
$$\mathbf{x}_k = \sum_{v_i \in d_k} \frac{tf(d_k, v_i)}{len(d_k)} \times \mathbf{t}_i. \quad (2)$$



**Fig. 2.** Las palabras de cada documento, previamente representadas en el espacio de perfiles (tabla izquierda), son utilizadas para representar al documento en el mismo espacio de perfiles (tabla derecha), el aporte de cada palabra esta dado por la ecuación 2. De esta forma se obtiene la representación de los documentos en el espacio de perfiles.

**Generación de subperfiles.** Una vez que se tiene la representación de los documentos en el espacio de perfiles se procede a hacer la búsqueda de *subperfiles*,

lo cual se logra en dos etapas. Primeramente se agrupan los documentos que pertenecen al mismo perfil, y luego se aplica algún algoritmo de agrupamiento a cada perfil, por ejemplo *k-means*, véase Figura 3. El problema que se tiene al usar *k-means* (y otros algoritmos de agrupamiento), es que requiere la cantidad específica de agrupaciones que debe encontrar, sin embargo, muchas veces este valor es desconocido, por lo que una forma de obtener un buena cantidad de agrupaciones es ejecutar el algoritmo de agrupamiento varias veces, indicando diferentes cantidades en cada ocasión y usando una medida de validación para encontrar la mejor agrupación. En este trabajo se usó el algoritmo de *k-means* con el coeficiente de *Silhouette* para encontrar las mejores agrupaciones por perfil, usando las implementaciones de *scikit learn*.



**Fig. 3.** Una vez representados los documentos en el espacio de perfiles, se aplican técnicas de agrupamiento para encontrar los subperfiles de cada perfil. Nótese que cada perfil puede tener diferente cantidad de subperfiles.

Los *subperfiles* encontrados (agrupaciones por perfil), serán vistos como las nuevas clases/perfiles, por ejemplo, en la Figura 3 se encontraron 7 *subperfiles*, de este modo se tendrían 7 clases/perfiles. El siguiente paso es repetir la *Representación de función de los perfiles* y posteriormente la *Representación de los Documentos*, por lo tanto, al final se obtiene una matriz de los documentos en el espacio de *subperfiles*, ver Figura 4.

	subp_1	subp_2	subp_3	subp_4	subp_5	subp_6	subp_7
d1	0.27	0.01	0.14	0.283	0.21	0.15	0.17
...	0.27	0.25	0.21	0.185	0.03	0.05	0.25
...	0.179	0.17	0.17	0.167	0.07	0.08	0.16
d <sub>n</sub>	0.008	0.14	0.04	0.024	0.36	0.34	0.08

**Fig. 4.** Representación de documentos en el espacio de subperfiles.

**Clasificación.** Una vez finalizada la representación de los documentos a *SOA2*, la matriz resultante está lista para entrenar a cualquier clasificador que tome como entrada una matriz de ejemplos por atributos más una lista de la clase a la que pertenece cada ejemplo. Para este trabajo se han usado 2 clasificadores, el Naïve Bayes y una máquina de soporte vectorial (SVM), ambos clasificadores de WEKA 3.6, los resultados pueden verse en la Sec. 4.

**Ventajas.** Algunas de las ventajas de esta técnica son las siguientes:

- **Reducción de dimensión:** Comparado con una bolsa de palabras (BoW), el tamaño de los vectores en general es mucho mas pequeño que los vectores de la BoW, ya que el tamaño de los vectores de una BoW esta dado por el tamaño del vocabulario.
- **Matriz NO dispersa:** ligado con el punto anterior, y uno de los problemas bien conocidos de la BoW, es que se obtienen matrices con datos dispersos, sin embargo, esto no sucede con *SOA2*.

**Desventajas.** Una de las posibles desventajas de este método es la dificultad para determinar la cantidad óptima de *superfiles* por clase, lo cual impacta en el rendimiento y eficacia del método.

### 3.2. Corpus y preprocesamiento

En este trabajo se han usado los siguientes cuatro corpus:

- **OpSpam [6, 5]:** Contiene 800 opiniones reales y 800 opiniones falsas acerca de hoteles situados en Chicago. Las opiniones verdaderas fueron extraídas de notas reales de *TripAdvisor*, mientras que las falsas fueron requeridas vía *Amazon Mechanical Turk* (AMT).
- **Temas controversiales (*Abortion, Death Penalty, Best Friend*) [4]:** En los 2 primeros temas, se pidió a algunas personas escribir su opinión (opiniones reales) y posteriormente se les pidió que escribieran una opinión contraria o lo que habían escrito previamente (opiniones falsas). De forma similar para el tercer tema, se pidió a algunas personas escribir sobre su mejor amigo (opiniones reales) y posteriormente se les pidió escribir sobre una persona que no soporten, como si fuera su mejor amigo (opinión falsa). Finalmente, se tienen 100 opiniones reales y 100 opiniones falsas por cada uno de los tres temas.

El preprocesamiento que se llevó a cabo en estos corpus fue reducción a minúsculas y la eliminación de signos de puntuación, enfocándose únicamente en las palabras. Para llevar a cabo la evaluación, se ha hecho con validación cruzada de 5 pliegues, es decir, 80 % para entrenamiento y 20 % para prueba por cada pliegue.



#### 4. Resultados

En las tablas 1, 2, 3 y 4 se presentan los resultados para los corpus *OpSpam*, *Abortion*, *Best Friend* y *Death Penalty*, respectivamente. Dado que se generaron 5 pliegues, los resultados mostrados en las tablas para la Exactitud (E), Precisión (P), Recuerdo (R) y la medida  $F_1$ , son calculadas con el macro-promedio. La mejor exactitud la consigue en el corpus de *OpSpam*, alcanzando un 83.9, y el peor resultado lo consigue en el corpus de *Death Penalty*, donde solo logra alcanzar una exactitud del 56.0 %.

**Tabla 1.** Resultados en *OpSpam*.

Clasif.	Máx.	Prom. Subperfiles		E	P		R		$F_1$	
	Subperfiles	F	V		F	V	F	V	F	V
NB	2	$2 \pm (0)$	$2 \pm (0)$	<b>83.9</b>	0.855	0.825	0.819	0.860	<b>0.836</b>	<b>0.842</b>
SVM				81.9	0.879	0.776	0.740	0.897	0.802	0.832
NB	5	$2 \pm (0)$	$2 \pm (0)$	<b>83.9</b>	0.855	0.825	0.819	0.860	<b>0.836</b>	<b>0.842</b>
SVM				81.8	<b>0.880</b>	0.775	0.737	<b>0.898</b>	0.801	0.831
NB	10	$3.6 \pm (3.57)$	$3.6 \pm (3.57)$	83.8	0.850	<b>0.828</b>	<b>0.824</b>	0.852	<b>0.836</b>	0.840
SVM				80.5	0.870	0.762	0.717	0.892	0.785	0.821

**Tabla 2.** Resultados en *Abortion*.

Clasif.	Máx.	Prom. Subperfiles		E	P		R		$F_1$	
	Subperfiles	F	V		F	V	F	V	F	V
NB	2	$2 \pm (0)$	$2 \pm (0)$	74.0	0.797	0.707	0.660	0.820	0.718	0.756
SVM				74.5	<b>0.803</b>	0.710	0.660	<b>0.830</b>	0.721	0.763
NB	5	$4 \pm (1.41)$	$4 \pm (1.41)$	<b>76.0</b>	<b>0.803</b>	<b>0.733</b>	<b>0.700</b>	0.820	<b>0.745</b>	<b>0.771</b>
SVM				74.0	0.792	0.706	0.660	0.820	0.718	0.757
NB	10	$6.8 \pm (3.96)$	$5.8 \pm (3.56)$	73.0	0.782	0.696	0.650	0.810	0.708	0.748
SVM				74.0	0.780	0.719	0.690	0.790	0.727	0.748

En general, la cantidad de *subperfiles* de cada clase (i. e.  $F$  y  $V$ ) tiene un comportamiento distinto entre las opiniones sobre hoteles y las controversiales. En particular, en las primeras el promedio de *subperfiles* es menor y con menor varianza, mientras que en las segundas parece aumentar proporcionalmente al parámetro de máximo agrupamiento, con excepción del corpus *Death Penalty*. Esto puede deberse tanto a la cantidad de datos que se tienen como a la forma en que cada corpus fue construido. Por ejemplo, para las opiniones verdaderas sobre hoteles, *TripAdvisor* insta a los usuarios a evaluar el hotel en función de aspectos específicos como localidad, limpieza, calidad del sueño, precios [2]. Sin embargo, las opiniones controversiales fueron adquiridas sin ningún tipo de restricción,

**Tabla 3.** Resultados de *Best Friend*.

Clasif.	Máx.	Prom. Subperfiles		E	P		R		$F_1$	
	Subperfiles	F	V		F	V	F	V	F	V
NB	2	$2\pm(0)$	$2\pm(0)$	78.0	0.833	0.768	0.740	0.820	0.771	0.779
SVM				<b>81.5</b>	0.899	0.772	0.720	0.910	0.791	<b>0.831</b>
NB	5	$3.2\pm(0.44)$	$3.6\pm(0.89)$	<b>81.5</b>	0.861	<b>0.793</b>	<b>0.760</b>	0.870	<b>0.800</b>	0.824
SVM				79.5	<b>0.911</b>	0.738	0.660	<b>0.930</b>	0.757	0.821
NB	10	$8.2\pm(2.68)$	$6.8\pm(3.27)$	80.5	0.871	0.774	0.730	0.880	0.784	0.819
SVM				78.0	0.858	0.735	0.670	0.890	0.747	0.803

**Tabla 4.** Resultados de *Death Penalty*.

Clasif.	Máx.	Prom. Subperfiles		E	P		R		$F_1$	
	Subperfiles	F	V		F	V	F	V	F	V
NB	2	$2\pm(0)$	$2\pm(0)$	55.5	0.557	0.556	<b>0.560</b>	0.550	<b>0.555</b>	0.550
SVM				55.0	0.562	0.543	0.46	0.64	0.502	0.585
NB	5	$2\pm(0)$	$2\pm(0)$	<b>56.0</b>	0.566	<b>0.557</b>	0.550	0.570	0.554	0.560
SVM				55.5	<b>0.570</b>	0.547	0.460	<b>0.650</b>	0.504	<b>0.591</b>
NB	10	$3.4\pm(3.13)$	$2.8\pm(1.78)$	44.5	0.557	0.536	0.504	0.580	0.512	0.551
SVM				54.5	0.554	0.540	0.450	0.640	0.492	0.582

por lo que los individuos tuvieron total libertad de expresarse en dichos temas pudiendo ser tan específicos o amplios según quisieran.

Para poner en contexto los resultados obtenidos, en la tabla 5 se comparan los resultados contra el método tradicional de bolsa de palabras (*BoW*), y otros métodos del estado del arte que no utilizan recursos externos o herramientas de análisis lingüístico.

**Tabla 5.** Comparación con el estado del arte

Corpus	Trabajos	Exactitud
<i>OpSpam</i>	<i>bigramas</i> <sup>+</sup> [1]	<b>86.0</b>
	<i>SOA2</i>	83.9
	<i>BoW</i>	84.5
<i>Abortion</i>	<i>unigramas</i> [3]	63.8
	<i>SOA2</i>	<b>76.0</b>
	<i>BoW</i>	69.5
<i>Best Friend</i>	<i>unigramas</i> [3]	74.5
	<i>SOA2</i>	<b>81.5</b>
	<i>BoW</i>	77.5
<i>Death Penalty</i>	<i>unigramas</i> [3]	58.1
	<i>SOA2</i>	56.0
	<i>BoW</i>	<b>62.5</b>

Como puede observarse los resultados obtenidos al aplicar *SOA2* son superiores en función del tipo de engaño. En el caso de *OpSpam* el método no muestra ventajas cayendo su rendimiento aún por debajo del método tradicional de *BoW*. En el caso de las controversiales, utilizar un enfoque basado en *subperfiles* tiene resultados alentadores, superando incluso el estado del arte en los corpus *Best Friend* y *Abortion*. Aún es necesario profundizar más en el análisis de estos resultados para determinar las diferencias de comportamiento entre los corpus de notas controversiales. No obstante, estos resultados sugieren que la hipótesis de que existen *subperfiles* de engañadores y veraces se debe tener en cuenta en futuros trabajos para la detección del engaño. Finalmente, los resultados de propuestas que emplean recursos externos [6] o reglas gramaticales [1] para la representación de los textos siguen estando más por encima de los alcanzados por los métodos simples. Así que en futuros trabajos se podría buscar estrategias que permitan incluir este tipo de información en las representaciones de *SOA2*, de forma que se considere esta información en la búsqueda de *subperfiles*.

## 5. Conclusiones y trabajo futuro

Se ha mostrado cómo el uso de la técnica *SOA2* [3], la cual crea una representación basada en *subperfiles*, ha alcanzado resultados interesantes, prueba de ello se puede apreciar en la tabla 5, donde se compararon los resultados con métodos del estado del arte. No obstante, el comportamiento difiere en función del tipo de engaño. En el caso de las valoraciones sobre hoteles el método no brinda ninguna ventaja, por el contrario con los corpus de notas controversiales se tienen resultados interesantes. Aún es necesario realizar un análisis más profundo para explicar este comportamiento el cual puede deberse tanto al tamaño de las colecciones, a la diversidad de tópicos y por ende al tamaño del vocabulario, etc.

Para el trabajo futuro se propone, por un lado: (i) retomar los subperfiles encontrados en el paso de Generación de Subperfiles, y agregarlos como subclases, lo cual daría un problema multidimensional, el cual podría ser tratado usando un enfoque de clasificadores encadenados, de modo que estas nuevas subclases aporten información relevante a la clase original. Alternativamente podría verse como un ensamble, donde se construye un clasificador para cada subclase, se evalúa el nuevo documento y la clase que tenga más votos será a la que pertenece (cada subclase pertenece en principio a una clase, por lo que el voto de la subclase es para la clase a la que pertenece); y por otro lado, (ii) dado que los métodos que usan recursos externos o información sintáctica han demostrado su potencial, se podría buscar estrategias que permitan incluir información de este tipo en la búsqueda de *subperfiles* dentro del *SOA2*.

**Agradecimientos** Este trabajo ha sido realizado con el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través de las becas No. 634411 y No. 613411, y del proyecto CONACYT CB-2015-01-257383.

## Referencias

1. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 497–501. Association for Computational Linguistics (2013), <http://www.aclweb.org/anthology/N13-1053>
2. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 309–319. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011), <http://dl.acm.org/citation.cfm?id=2002472.2002512>
3. Pérez-Rosas, V., Mihalcea, R.: Cross-cultural deception detection. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 440–445. Association for Computational Linguistics (2014), <http://www.aclweb.org/anthology/P14-2072>
4. Rosso, P., Cagnina, L.C.: Deception Detection and Opinion Spam, pp. 155–171. Springer International Publishing, Cham (2017), [https://doi.org/10.1007/978-3-319-55394-8\\_8](https://doi.org/10.1007/978-3-319-55394-8_8)

# Evaluación de candidatos para la retroalimentación de corpus por medio de bagging

Cecilia Reyes Peña, David Pinto Avendaño, Darnes Vilariño Ayala

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación, Puebla,  
México

reyesp.cecilia@gmail.com, {dpinto,darnes}@cs.buap.mx

**Resumen.** Las redes sociales contienen suficiente elementos para construir corpus acerca de temas novedosos, en dichos corpus existe la posibilidad de que se conviertan en obsoletos debido a la naturaleza efímera de dicha información. La tarea de retroalimentar corpus para mantenerlos vigentes es muy importante y a su vez, una tarea muy difícil cuando se hace manualmente debido a la cantidad de información que se tiene que manejar. En este trabajo se presenta la comparativa de los resultados de la aplicación de la técnica de ensamble de clasificadores Bagging con votación simple para la selección de Tweets candidatos con la finalidad de retroalimentar el corpus de entrenamiento. Dicho corpus está balanceado y dividido en cuatro clases (alegría, tristeza, ira y miedo) y el proceso de clasificación es realizado por medio de tres modelos: Ranking, Naïve Bayes y Probabilidad de Bigramas. Los candidatos son seleccionados de un conjunto de prueba etiquetado manualmente y la retroalimentación del corpus será evaluada por medio de pruebas K-Fold Cross Validation.

**Palabras clave:** bagging, ranking, Naïve Bayes, probabilidad de bigramas, retroalimentación de corpus, ensamble de clasificadores.

## Candidates Evaluation for Corpus Feedback by Bagging

**Abstract.** The social networks contains enough elements for build corpus about novel topics in which there is the possibility of become obsolete because to their fleeting nature. The feedback corpus task for keeping them current is very important and in turn a difficult and cost task because the used information amount for this. In this work, the comparative among the results of classifier ensemble technique with simple voting in order to candidates Tweets selection for train corpus feedback has been presented. The train corpus is divided in four classes (happiness, sadness, anger and fear) and the classification process is performed by three models: Ranking, Naïve Bayes and Bigrams Probabilities. The

candidates are selected from a manually tagged test set and the feedback is evaluated by K-Fold Cross Validation.

**Keywords:** bagging, ranking, Naïve Bayes, bigrams probability, corpus feedback, classifier ensemble.

## 1. Introducción

Las redes sociales como Twitter contienen una gran cantidad de información cambiante, es decir, los temas manejados dentro de estas son recientes y solo se mantienen vigentes por cortos lapsos de tiempo. Al trabajar con corpus de dicha información es necesario mantenerlos actualizados, lo cual es un proceso muy difícil y costoso si se hace manualmente. Existen técnicas como Bootstrapping para retroalimentar corpus de forma dinámica, por medio de un proceso de clasificación de información a base de etiquetado partiendo de lo particular a lo general, es decir parte de un elemento semilla para extraer un conjunto mayor de información que será clasificado mediante uno o varios modelos de clasificación [1]. Para que esto sea posible es necesario implementar estrategias de selección de elementos candidatos por medio de un clasificador, el cual considera algunas características y desprecia otras. Una forma de aprovechar múltiples características de la información es a través de la implementación de múltiples clasificadores que puedan aportar resultados según su naturaleza, siendo esta una de las principales ventajas del ensamble de clasificadores.

Se puede definir al ensamble de clasificadores como un trabajo colaborativo entre diversos modelos de clasificación dentro de una misma tarea, de tal forma que dicha colaboración ofrezca mejores resultados respecto a los que puedan otorgar cada uno de los clasificadores participantes por separado, simulando la naturaleza humana de pedir opiniones [2]. Existen diferentes técnicas de ensamble y entre las más populares podemos mencionar al Bagging, en el cual se generan múltiples versiones de una predicción de clase, la predicción final será definida mediante un proceso de votación. En términos generales, las diferentes versión se crean a partir de la agregación de elementos tomados al azar del conjunto de entrenamiento para generar nuevos conjuntos de entrenamiento que varían en cada iteración obteniendo así las versiones de la predicción [3]. Otra técnica popular es el boosting en el cual se le asigna peso a cada ejemplo y en cada iteración se modifica, al finalizar las iteraciones se realiza una votación de los resultados considerando los pesos finales [10].

En este trabajo se compara los resultados de la técnica bagging aplicada a un proceso de selección de tweets candidatos para la retraining de un corpus etiquetado manualmente en cuatro clases (alegría, tristeza, ira y miedo), mediante tres modelos de clasificación: uno basado en la técnica de recuperación de la información Ranking, Naïve Bayes y Probabilidad de bigramas.

Este documento está dividido en las siguientes secciones: la sección 2 contiene los trabajos relacionados al ensamble de clasificadores, la sección 3 describe los clasificadores utilizados en este trabajo, la sección 4 contiene los resultados de

la evaluación de los candidatos proporcionados por el ensamble, por último la sección 5 contiene las conclusiones de este trabajo.

## **2. Trabajos relacionados**

Dentro del campo del procesamiento del lenguaje natural, la técnica de ensamble de clasificadores ha sido utilizada en diversos proyectos, uno de ellos es la mejora de procesos de análisis de opinión afectiva, mediante elementos morfosintácticos para la búsqueda de relaciones avanzadas difíciles de detectar, utilizando algoritmos de aproximación simple, basado en DAL (Diccionario de Afecto en el Lenguaje), basado en otro diccionario de mayores dimensiones y un árbol de sintáxis con información morfológica. Para la clasificación se utiliza el algoritmo MaxEnt (Máxima Entropía) y para el ensamble se utiliza la técnica bagging con muestras del 80 % para entrenar y los resultados son analizados en un nuevo clasificador [8]. Otra aplicación del ensamble es durante la extracción de características de rostro y audio por medio de los algoritmos PCA-engenfaces y MFCC respectivamente, los clasificadores utilizados fueron Redes Bayesianas, Naïve Bayes, K-vecinos cercanos, Redes Neuronales y Árboles de decisión; para el ensamble entre ellos se utilizó la técnica Staking, en la cual los clasificadores tienen una jerarquía y la predicción de los clasificadores mas altos influye en los mas bajos [9]. También se ha aplicado el ensamble en la identificación de correos electrónicos denominada spam, donde los datos utilizados fueron extraídos de corpus públicos considerando de estos solo el contenido que el usuario puede leer representado en vectores de relevancia, para la clasificación se utilizaron los modelos: SVM (Support Vector Machine), Árbol de decisión, Red Neuronal, K vecinos cercanos, Naïve Bayes, entre otros. El ensamble se realizó mediante la técnica Boosting apoyando la decisión por medio de votos simples [11]. GuoDong Zhou et al. [12] presenta un modelo de reconocimiento de nombres de genes y proteínas en texto considerando también sus abreviaturas por medio de un módulo donde cada término sea emparejado con un diccionario. Los modelos de clasificación utilizados son: SVM y dos modelos ocultos de Markov discriminativos, dejando la decisión final a una votación mayoritaria simple. Por su parte, Onan Aytuğ et al. [13] presenta la comparativa de varias técnicas de ensamble como AdaBoost, Bagging, Dagging, Random Subspace y Votación mayoritaria simple; en la tarea de extracción de palabras clave usando los modelos de clasificación Naïve Bayes, SVM, Regresión Logística y Random Forest. Los mejores resultados de esta comparativa fueron los otorgados por la técnica Bagging aplicada al clasificador Random Forest.

## **3. Clasificadores**

En esta sección se describen los clasificadores utilizados para la técnica del ensamble en este trabajo.

### 3.1. Ranking

El modelo de clasificación utilizado en este trabajo está basado en la técnica de recuperación de la información Ranking simple, donde se aplican algoritmos para crear listas de relevancia basándose en las características propias de una colección de documentos [4]. Para esto se crearon cuatro documentos (uno por clase) para formar un corpus o colección, posteriormente es necesario conocer la frecuencia de cada palabra dentro de cada documento ( $TF_{t,d}$ ) y la relevancia que tiene la misma dentro del corpus ( $IDF_t$ , Inverse Document Frequency of the Term), para esta última se emplea la Eq.1, donde  $N$  es el total de documentos del corpus,  $DF_t$  (Document Frequency of the Term) el número de documentos en el que aparece la palabra y se le aplica el logaritmo base diez para suavizar la relevancia. Para obtener el valor de relevancia final se utiliza la ecuación 2:

$$IDF_t = \log_{10}(N/DF_t), \quad (1)$$

$$TF - IDF_{t,d} = TF_{t,d} \times IDF_t. \quad (2)$$

El proceso para clasificar un tweet nuevo es similar a realizar una consulta, primero se hace un pesado de cada una de las palabras que lo conforman considerando la frecuencia de la palabra en el tweet  $TF_{t,q}$  (Term Frequency in the Query) y en el  $IDF_t$  de cada documento (Eq. 3), obteniendo así un vector compuesto por el  $TF - IDF$  de cada palabra. Finalmente, se realiza un producto punto del vector y el documento que aporte mayor puntaje es al que corresponde el tweet:

$$w_{t,q} = \log_{10}(1 + TF_{t,q} \times IDF_t). \quad (3)$$

Una de las modificaciones del algoritmo de Ranking propuesto en este trabajo es el cálculo de la relevancia de bigramas en un documento. Desde la lectura de los documentos del corpus de entrenamiento, el modelo agrupa bigramas de palabras consecutivas y los toma en cuenta como una sola palabra, al igual que en los tweets a clasificar, el modelo creará los bigramas del tweet y serán evaluados respecto a los almacenados (ver Figura 1).

### 3.2. Naïve Bayes

Naïve Bayes es uno de los clasificadores estadísticos supervisados más populares. Está basado en el Teorema de Bayes y asume que una característica de una clase en particular contribuye de forma independiente a la probabilidad de que cualquier otra característica pertenezca o no a la misma clase [5]. El funcionamiento del clasificador radica en calcular la probabilidad de pertenencia de cada característica  $X$  en cada una de las clases  $C$ , cuando se encuentra el valor más alto se retorna el nombre de la clase al cual pertenece el objeto (Ec. 4):

$$\hat{y} = \underset{k \in \{1, \dots, K\}}{\operatorname{argmax}} P(C) \prod_{i=1}^n P(x_i | C_K). \quad (4)$$



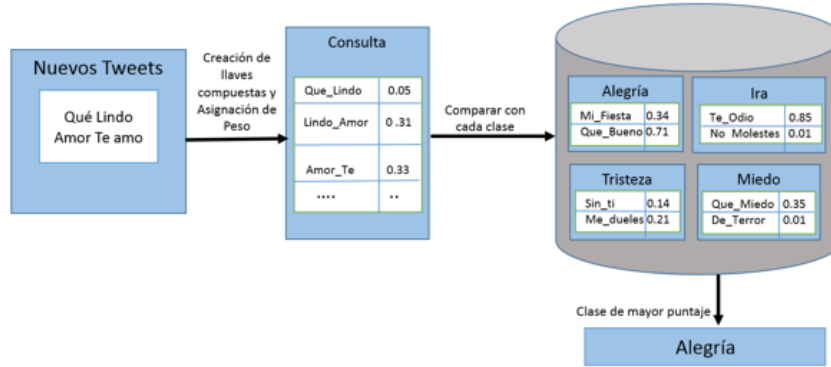


Fig. 1. Modelo Ranking con llaves compuestas.

### 3.3. Probabilidad de bigramas

Se puede definir al  $N$ -grama como una serie de  $N$  palabras consecutivas que conforman a una sentencia u oración y se denominan según su grado (el tamaño de la  $N$ ). En el procesamiento del lenguaje natural, los modelos más usados basados en  $N$ -gramas, son comúnmente unigramas (de una palabra), bigramas (de dos palabras) y trigramas (de tres palabras) [6]. Cabe resaltar que entre más grande sea el  $N$ -gramas contiene más información que el grado  $N-1$ . Para el cálculo de probabilidades en modelos basados en  $N$ -gramas, es frecuentemente utilizada la suposición de Markov, en la cual se asume que la probabilidad de una palabra depende solamente de las  $N-1$  palabras anteriores [7]. Por lo tanto, en un modelo de bigramas, la probabilidad de una sentencia ( $P(s)$ ) conformada por  $N$  palabras ( $w_1, w_2, \dots, w_n$ ) está dada por la multiplicación de las probabilidades de cada palabra dada la anterior (Ec. 5):

$$P(s) = P(w_1)P(w_2 | w_1) \dots P(w_n | w_{n-1}). \quad (5)$$

## 4. Resultados

Para la realización de las pruebas se utilizó un corpus de entrenamiento conformado por 105,598 tweets repartidos en forma balanceada en cuatro clases, al evaluarlo mediante k-Fold Cross Validation con una  $K=10$ , dando como resultado un porcentaje promedio de aciertos del 90 %. Por su parte, el conjunto de prueba está conformado por 234 tweets (127 de alegría, 31 de tristeza, 62 de ira y 14 de miedo), ambos etiquetados manualmente. Inicialmente se midió el rendimiento de cada clasificador respecto a aciertos y errores en la clasificación por medio de pruebas Precisión, Recall y F-Measure (ver Tabla 1).

En la selección de candidatos del conjunto de prueba, utilizando el corpus de entrenamiento original en los tres modelos de clasificación se obtuvieron los siguientes resultados (ver Tabla 2).

Se realizaron cinco pruebas utilizando el conjunto de prueba y entrenando con Bagging, tomando al azar muestras equivalentes al 80 % del tamaño del corpus de entrenamiento original para cada uno de los tres modelos. En la Tabla 3 se muestra el promedio de candidatos obtenidos para cada clase. Una vez retroalimentado el corpus de entrenamiento, se aplicó la prueba K-Fold Cross Validation para medir la integridad del corpus con un  $k=5$  (ver Tabla 4).

**Tabla 1.** Resultado de Medidas Presicion, Recall y F-Measure de los tres modelos de clasificación.

Model	Measure	Alegría	Tristeza	Ira	Miedo
Ranking	Precision	0.733	0.4	0.71	0.174
	Recall	0.693	0.387	0.435	0.571
	F-measure	0.712	0.393	0.53	0.266
Naïve Bayes	Precision	0.95	0.195	0.586	0.163
	Recall	0.299	0.548	0.435	0.714
	F-measure	0.455	0.288	0.5	0.266
Bigrams Probabilities	Precision	0.901	0.351	0.396	0.239
	Recall	0.574	0.612	0.338	0.785
	F-measure	0.701	0.447	0.365	0.366

**Tabla 2.** Número de candidatos por clase.

Emoción	Votación Mayoritaria	Votación Unánime
Alegría	89	25
Tristeza	43	11
Ira	33	14
Miedo	39	13
Total	204	63

## 5. Conclusiones y trabajo futuro

En este trabajo se presentó la comparación del rendimiento de la selección de tweets candidatos para la retroalimentación de corpus mediante la técnica de ensamble de clasificadores Bagging utilizando tres modelos de clasificación: Ranking, Naive Bayes y Probabilidad de bigramas.

El valor de las medidas Presicion, Recall y F-measure de los modelos Ranking y Probabilidad de bigramas son cercanas entre si, mientras que las medidas

**Tabla 3.** Promedio de candidatos por clase.

Emoción	Promedio Votación Mayoritaria	Promedio Votación Unánime
Alegría	75	26
Tristeza	46	9
Ira	40	19
Miedo	37	13
Total	198	67

**Tabla 4.** Validación de corpus retroalimentado.

Iteración	Retroalimentación por votación mayoritaria			Retroalimentación por votación unánime		
	Ranking	Naïve Bayes	Probabilidad de Bigramas	Ranking	Naïve Bayes	Probabilidad de Bigramas
1	0.739	0.8	0.773	0.788	0.852	0.826
2	0.838	0.904	0.887	0.83	0.896	0.878
3	0.715	0.822	0.777	0.768	0.886	0.819
4	0.761	0.865	0.805	0.762	0.866	0.805
5	0.795	0.863	0.833	0.826	0.882	0.855
Promedio	0.769	0.851	0.815	0.795	0.876	0.836

de Naïve Bayes son mas bajas respecto a las dos anteriores. Los candidatos obtenidos por la votación mayoritaria no son confiables como se desearía puesto que sugiere una mayor cantidad de candidatos respecto a los que se tienen etiquetados en el conjunto de prueba, tal es el caso para las clases tristeza y miedo.

En la prueba de selección de candidatos donde se utiliza el corpus de entrenamiento original para los tres modelos, la cantidad de elementos seleccionados está dentro del rango del porcentaje de las pruebas que se realizaron tomando segmentos aleatorios del corpus de entrenamiento para cada modelo, por lo que se asume que los rendimientos de los clasificadores en ambos casos es similar.

Los resultados obtenidos de la validación de corpus muestran que el corpus que fue retroalimentado con los candidatos sugeridos por votación mayoritaria se vio más afectado en la integridad respecto al que fue retroalimentado con los candidatos sugeridos por votación unánime, a pesar de esto, ambos presentan una baja en cuanto a la validación original que fue del 90 %.

Como trabajo futuro, se realizarán más pruebas de retroalimentación para visualizar las modificaciones que pueda sufrir el corpus, además se implementará la técnica Bootstrapping para la complementar una retroalimentación automática del corpus integrando modelos de selección de elementos semilla para la extracción masiva de tweets que puedan ser clasificados por los modelos aquí presentados.

## Referencias

1. Enríquez, F.: Técnicas de Bootstrapping en el Procesamiento del Lenguaje Natural. Memoria del Periodo de Investigación en el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla (2007)
2. Rokach, L.: Pattern Classification Using Ensemble Methods. Series in machine perception and artificial intelligence, 75, World Scientific (2010)
3. Breiman, L.: Bagging predictors. Machine learning, 24(2), pp. 123–140 (1996)
4. Li, H.: Learning to rank for information retrieval and natural language processing. Synthesis Lectures on Human Language Technologies, 7(3), pp. 1-121 (2014)
5. Russell, S., Norving, P.: Artificial Intelligent A Modern Approach. Third Edition, Prentice Hall (2010)
6. Rodríguez, H.: Lingüística y estadística, incompatibles?. Tecnologías del texto y del habla, 72(89) (2004)
7. Basharin, G.P., Langville, A.N., Naumov, V.A.: The life and work of AA Markov. Linear algebra and its applications, 386, pp. 3–26 (2004)
8. De la Vega, M., Vea-Murguía, J.: Ensemble algorithm with syntactical tree features to improve the opinion analysis. Comité organizador, 53 (2015)
9. Carrasco, A., Portugal, R., Peralta, B.: Reconocimiento biométrico de audio y rostro: un sistema viable de identificación. Pontificia Universidad Católica de Chile Departamento de Ciencia de la Computación (2006)
10. Cadenas, J.M., Garrido, M.C., Díaz, R.A.: Soft computing en ensambles basados en boosting, bagging y random forest. Rev. Iberoam. de Sistemas, Cibern. e Informática, 7(1), pp. 25-32 (2010)
11. Hernández, J., Higuera, O., Martínez-Trinidad, J.: Detección de correo electrónico Spam usando clasificadores supervisados. Seminario Nacional de Aprendizaje e Inteligencia Computacional (2014)
12. Zhou, G., Shen, D., Zhang, J., Su, J., Tan, S.: Recognition of protein/gene names from text using an ensemble of classifiers. BMC bioinformatics, 6(1), S7 (2005)
13. Onan, A., Korukoğlu, S., Bulut, H.: Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications, 5(7), pp. 232–247 (2016)

## **Interfaz de lenguaje natural para consultar cubos multidimensionales utilizando procesamiento analítico en línea**

J. A. Porras Medrano, R. Florencia-Juárez, G. Rivera Zárate, V. García Jiménez

Universidad Autónoma de Ciudad Juárez, Chihuahua,  
México

al133614@alumnos.uacj.mx,  
{rogelio.florencia, gilberto.rivera, vicente.jimenez}@uacj.mx

**Resumen.** El Procesamiento Analítico en Línea (On-Line Analytical Processing, u OLAP, por sus siglas en inglés) es una solución en el área de Inteligencia de Negocios que emplea estructuras multidimensionales, es decir, cubos OLAP, con el fin de agilizar el procesamiento y el acceso a grandes volúmenes de datos, normalmente almacenados en un almacén de datos (DW). Para acceder a los datos almacenados en un cubo, en muchas ocasiones se requiere que los usuarios formulen consultas en el lenguaje MultiDimensional eXpressions (MDX), conocimiento técnico que la mayoría de los usuarios no posee. En este trabajo se describe la implementación de una Interfaz de Lenguaje Natural (ILN) cuyo objetivo es traducir una consulta formulada en lenguaje natural a una consulta MDX. El conocimiento que la ILN requiere para efectuar la traducción es modelado semánticamente a partir de los distintos elementos que componen la estructura de los cubos. El modelado es realizado automáticamente por un módulo de configuración, el cual utiliza representaciones semánticas diseñadas en el Lenguaje de Ontologías Web (Web Ontology Language, OWL por sus siglas en inglés).

**Palabras clave:** interfaz de lenguaje natural, conocimiento semántico, consulta MDX, cubos OLAP, base de datos multidimensionales.

### **Natural Language Interface for Querying Multidimensional Cubes by using On-Line Analytical Processing**

**Abstract.** On-Line Analytical Processing (OLAP) is a solution in the Business Intelligence field that uses multidimensional structures, that is to say, OLAP cubes, with the aim of speed up processing and access to large volumes of data, usually stored in a data warehouse (DW). In order to access the data stored in a cube, it is very often for users to formulate queries in the MultiDimensional eXpressions language, but that is technical knowledge which most of them do not possess. This work describes the implementation of a Natural Language Interface (NLI) which main objective is to translate a query formulated in natural language to an MDX query. The knowledge required for the NLI to perform the translation process is modeled semantically from the different elements that

compose the structure of the cubes. Modeling is performed automatically by a configuration module, which uses semantic representations designed in the Ontologies Web Language (OWL).

**Keywords:** natural language interface, semantic knowledge, MDX query, OLAP cubes, multidimensional databases.

## **1. Introducción**

El área de Inteligencia de Negocios está enfocada en transformar datos provenientes de sistemas de gestión empresarial en conocimiento, con lo cual se pueda optimizar el proceso de toma de decisiones estratégicas. Para lograr esto se han desarrollado numerosas aplicaciones y herramientas de software que permiten obtener información significativa de fuentes de información y datos, tales como bases de datos y almacenes de datos (DataWarehouse, DW por sus siglas en inglés).

En algunas ocasiones, realizar tal labor requiere normalmente de cierto grado de conocimientos técnicos de lenguajes de consultas como SQL (Structured Query Language) o MDX (MultiDimensional eXpressions), obligando a usuarios convencionales a recurrir a expertos con el fin de recuperar información de las fuentes mencionadas.

Para facilitar a los usuarios este proceso, investigadores se han enfocado en diseñar interfaces que permitan traducir automáticamente una consulta expresada en lenguaje natural por los usuarios a una consulta MDX.

En este trabajo se propone la arquitectura de una interfaz de lenguaje natural (ILN) capaz de traducir de una consulta formulada por el usuario en el idioma inglés a una consulta MDX, la cual se utiliza para extraer información de cubos multidimensionales, almacenados en un DW. El funcionamiento se divide principalmente en dos módulos, el Módulo Generador de Conocimiento (MGC) y el Módulo de la Interfaz (MI). El MGC se encarga de modelar semánticamente el conocimiento que el MI utiliza para interpretar la consulta del usuario y generar la consulta MDX correspondiente.

El modelado se genera automáticamente a partir de la estructura de los cubos de un DW, utilizando representaciones semánticas diseñadas en el Lenguaje de Ontologías Web (Web Ontology Language, u OWL, por sus siglas en inglés). En la Sección 2 se presentan trabajos relacionados; en la Sección 3 se presenta la arquitectura de la interfaz propuesta; en la Sección 4 se presenta una discusión acerca del desempeño de la interfaz y, en la Sección 5 se presentan conclusiones y trabajos futuros.

## **2. Trabajos relacionados**

En el trabajo de Kuchmann-Beauger et al. [1] se propone una ILN capaz de procesar preguntas expresadas por los usuarios en lenguaje natural. Su enfoque consiste en identificar palabras clave o entidades en la expresión ingresada por el usuario y enlazar dichas entidades a objetos definidos previamente en un modelo de datos. Su objetivo fue demostrar que reescribir la consulta del usuario a través del uso de un tesoro cuando no se ha encontrado ninguna respuesta, produce mejores resultados. Los elementos semánticos identificados en la pregunta del usuario se traducen a una

consulta MDX que el sistema pueda entender. Su ILN contiene un grafo de objetos y restricciones descritos en el modelo del DW.

Si bien la propuesta muestra resultados prometedores, al mismo tiempo los autores reconocen que el sistema aún podría ser más preciso si se convierte el tesoro en una ontología que ya no solo permita reescribir la consulta del usuario, sino además recolectar información de fuentes no estructuradas para validar las respuestas proporcionadas.

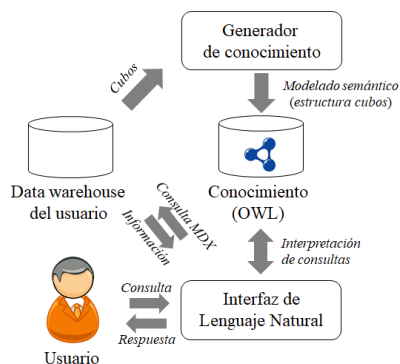
Por otro lado, en el trabajo de Saias et al. [2] se desarrolló BINLI, una ILN enfocada al Procesamiento Analítico En Línea (OnLine Analytical Processing, OLAP por sus siglas en inglés) cuyo funcionamiento se basa en una ontología. Las herramientas OLAP son soluciones para análisis de datos multidimensionales que permiten al usuario controlar la perspectiva y grado de detalle en cada dimensión del análisis en grandes volúmenes de datos.

BINLI pretende simplificar y hacer más flexible e intuitiva la interacción de usuarios con herramientas OLAP permitiéndoles realizar consultas en NL. A diferencia del trabajo de Kuchmann-Beauger et al., cuya ILN tiene un tesoro como base de conocimiento, BINLI se encuentra basado en una ontología, permitiendo inferencia en procesos de complejidad semántica mayores. BINLI utiliza análisis gramatical, reconocimiento de entidades, análisis morfosintáctico, cálculo de similitud semántica y razonamiento semántico. Cuenta con un generador de consultas OLAP MDX que permite evaluar varias interpretaciones de una misma consulta ordenadas descendientemente por orden de peso. La interpretación de mayor peso es enviada a un motor OLAP para su ejecución.

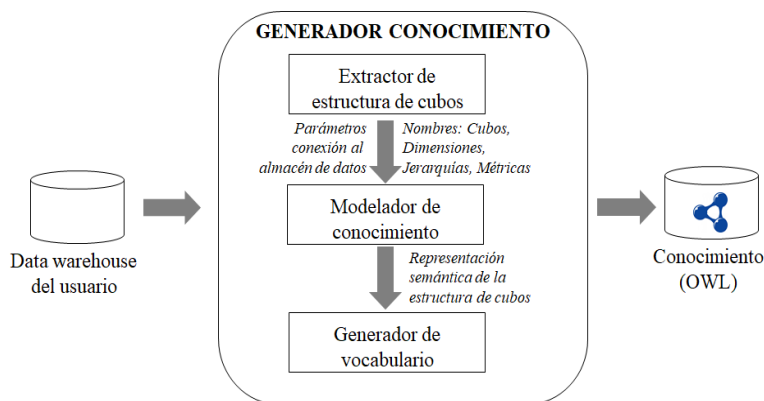
Como motor OLAP utilizan Pentaho Analysis Services Community Edition. BINLI también detecta errores de escritura utilizando la distancia Levenshtein. Además, detecta relaciones indirectas probando la compatibilidad semántica entre los términos a través del uso de una ontología de apoyo.

La propuesta de Prat, Akoka et al. [3] presenta un enfoque que busca definir una ontología basada en lógica descriptiva (OWL-DL) a partir de un modelo multidimensional. Las dimensiones, jerarquías de las dimensiones, niveles de la dimensión de la jerarquía, atributos del nivel de la dimensión, rollups, hechos, mediciones, funciones de agregación y tipos de agregación del modelo dimensional son representados en una ontología OWL-DL. Con el uso de la ontología OWL-DL demostraron facilitar la inferencia de la información deseada por el usuario, además de permitir una representación concisa y formal del conocimiento.

Configurar una ILN para una base de datos es un trabajo considerable, por tal motivo, Popescu et al. [4] decidieron implementar una ILN llamada PRECISE con la inclusión de un parser (analizador gramatical) estadístico. Implementaron un modelo semántico robusto para corregir errores de análisis gramatical y un marco de trabajo teórico para diferencia entre preguntas manejables y difíciles. Entrenaron el parser utilizando un conjunto de 150 preguntas donde cada palabra es etiquetada con etiquetas *Parte del Discurso* (Part of Speech, PoS por sus siglas en inglés) [5]. Utilizaron el parser Charniak como base para sus experimentos. PRECISE puede corregir errores sintácticos, complementos de preposición y elipsis de preposición. Cuando una decisión del parser resulta inconsistente con la información semántica del léxico de PRECISE, este último intenta reparar el árbol de análisis gramatical del parser. No obstante, PRECISE no puede realizar este tipo de correcciones en ciertos problemas



**Fig. 1.** Módulos de la interfaz.



**Fig. 2.** Arquitectura del Módulo Generador de Conocimiento.

como acoplamiento de verbos, de cláusulas, frases numéricas de sustantivos, entre otros, obligando al usuario a reformular su pregunta.

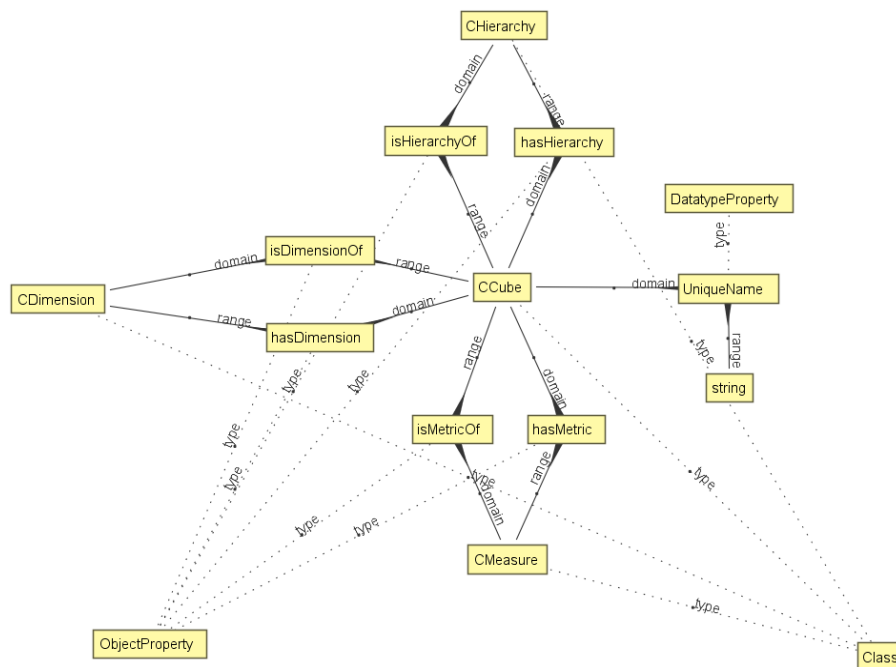
### 3. Arquitectura propuesta

La ILN propuesta en este artículo requiere de conocimiento del dominio, es decir, requiere conocer la estructura de los cubos multidimensionales almacenados en el DW con el fin de interpretar las consultas en lenguaje natural y generar las consultas MDX correspondientes. Por tal motivo, el diseño de la ILN se dividió en dos módulos, el *Módulo Generador de Conocimiento* (MGC) y el *Módulo de la Interfaz* (MI). En la Fig. 1 se presentan ambos módulos. En la Sección 3.1 se describe el MGC y en la Sección 3.2 se describe el MI.

#### 3.1. Módulo generador de conocimiento

Como se pudo ver en la Fig. 1, el objetivo del MGC es generar el conocimiento del MI. Para este fin, se analiza la estructura de cada uno de los cubos multidimensionales





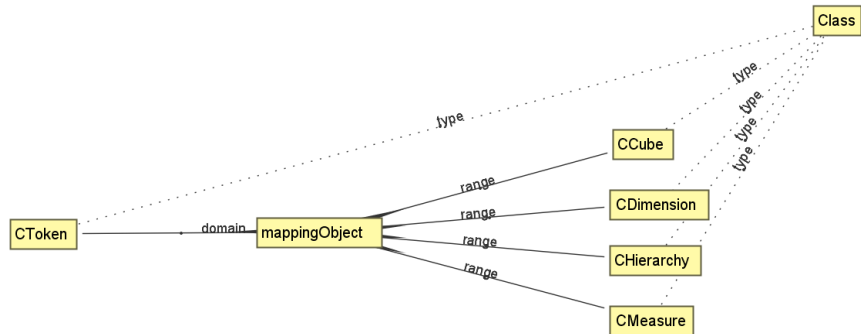
**Fig. 3.** Estructura de la representación semántica diseñada para modelar la estructura de los cubos.

almacenados en el DW con el objetivo de identificar los elementos que la componen (nombres de los cubos, dimensiones, jerarquía y métricas). Posteriormente, los elementos identificados son modelados semánticamente de manera automática utilizando una representación semántica que fue diseñada para este propósito.

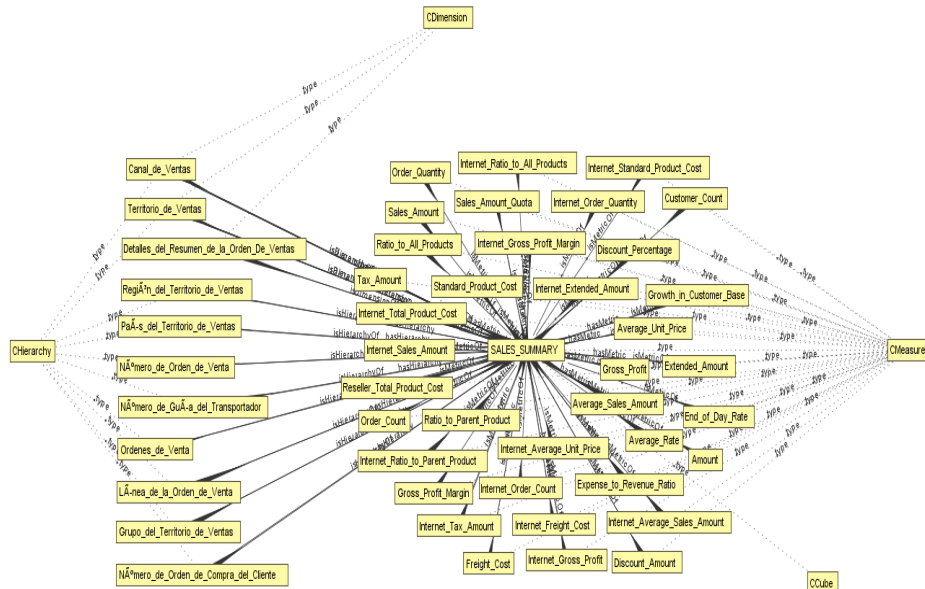
Por último, el modelado semántico es almacenado en una ontología, es decir, un archivo con formato XML y extensión OWL el cual constituye el conocimiento del MI. En la Fig. 2 se puede observar la arquitectura del MGC.

Como se puede apreciar en la Fig. 2, el MGC recibe como entrada los parámetros de conexión al DW. Mediante estos parámetros se establece una conexión con el DW y el *Extractor de estructura de cubos* analiza cada uno de los cubos del DW y extrae sus nombres, dimensiones, jerarquías y métricas. Cabe mencionar que sólo se extraen elementos que conforman la estructura de un cubo y no los datos que éste almacena. Los elementos extraídos son enviados al *Modelador de conocimiento*, el cual utiliza una representación semántica que fue diseñada para representar estos elementos.

La estructura de la representación semántica diseñada consta de clases (*CCube*, *CDimension*, *CHierarchy*, *CMeasure*), propiedades de objeto (*hasDimension*, *isDimensionOf*, *hasHierarchy*, *isHierarchyOf*, *hasMeasure*, *isMeasureOf*) y propiedades de datos (*UniqueName* de tipo *string*). Cada nombre de cubo es modelado como un individuo instanciado a partir de la clase *CCube*. Cada nombre de las dimensiones de un cubo es modelado como un individuo de la clase *CDimension* y es relacionado al individuo *CCube* al que pertenece mediante las propiedades de objeto *isDimensionOf* y *hasDimension* y así sucesivamente con los demás elementos



**Fig. 4.** Estructura de las representaciones semánticas diseñadas para modelar el vocabulario y su relación con los elementos del modelado semántico.



**Fig. 5.** Ejemplo del conocimiento modelado y almacenado en una de las ontologías generadas.

identificados. En la Fig. 3 se presenta la estructura de la representación semántica diseñada.

Después de haber modelado semánticamente la estructura de los cubos, el modelado es enviado al *Generador de vocabulario* cuyo objetivo es construir el vocabulario inicial del MI, el cual se compone de todos los nombres de cada uno de los elementos modelados.

Posteriormente, para incrementar la cobertura lingüística, cada palabra en el vocabulario es adicionada con sinónimos, así como con palabras que compartan el mismo lema.

Finalmente, todas las palabras son agregadas al modelado semántico utilizando una representación semántica diseñada para este fin. Esta representación consta de una clase llamada *CToken* y de las propiedades de objeto *mappingObject* e *isMappedBy*. Cada palabra es modelada como un individuo de la clase *CToken* y este individuo es relacionado a través de la propiedad *mappingObject* a los elementos a los que éste mapea, es decir, a los elementos de los que fue generado.

Es importante mencionar que este proceso es esencial para que el MI pueda interpretar las consultas de los usuarios. En la Fig. 4 se puede observar la estructura de la representación semántica diseñada para modelar el vocabulario y su relación con los elementos del modelado semántico.

Posterior a los procesos anteriormente mencionados, se genera la ontología (archivo de extensión.owl) en la que se va a almacenar el modelado semántico generado por el MGC.

En la Fig. 5 se presenta un fragmento del conocimiento modelado a partir del cubo SALES SUMMARY de la base de datos AdventureWorksDW2014, la cual es una base de datos de ejemplo de Microsoft. En la Fig. 5 se pueden ver las clases *CCube*, *CDimension*, *CHierarchy* y *CMeasure*, las cuales identifican las dimensiones, las jerarquías, las métricas y el nombre del cubo.

Los nombres mostrados son instancias de estas clases. Cabe mencionar que para facilitar la interpretación de la Fig. 5, no se muestran los respectivos sinónimos y palabras que comparten el mismo lema que los nombres de las instancias de estas clases.

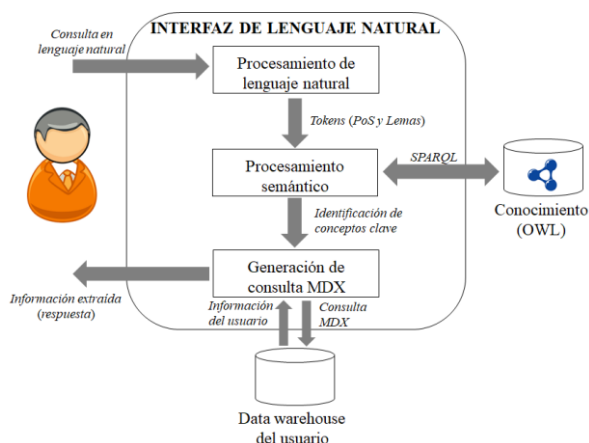


Fig. 6. Arquitectura del Módulo de la Interfaz.

### 3.2. Módulo de la interfaz

El objetivo del MI es traducir la consulta formulada en lenguaje natural por el usuario a una consulta MDX. Como primer paso se realiza un *procesamiento de lenguaje natural* a la consulta introducida por el usuario. Posteriormente se realiza un

**Tabla 1.** Procesamiento de lenguaje natural realizado por la herramienta Freeling.

TOKEN	LEMA	POS	SIGNIFICADO POS
Show	show	VB	pos=verb vform=infinitive
Me	me	PRP	pos=pronoun type=personal
Internet	internet	NP	pos=noun type=proper
sales	sales	NNS	pos=noun type=common num=plural
amount	amount	NN	pos=noun type=common num=singular
As	as	IN	pos=preposition
Per	per	IN	pos=preposition
customer	customer	NN	pos=noun type=common num=singular
.	.	FP	pos=punctuation type=period

*procesamiento semántico* con el fin de identificar sin ambigüedad conceptos clave en la consulta, así como los elementos de la ontología a los que hacen referencia. Finalmente, en base a los elementos identificados, se realiza el proceso de *generación de la consulta MDX*. En la Fig. 6 se puede apreciar la arquitectura del MI.

### 3.3. Procesamiento de lenguaje natural

Este paso consiste en analizar la consulta introducida por el usuario utilizando Freeling [6], la cual es una suite de herramientas utilizadas en el procesamiento de lenguaje natural que ofrece soporte para diversos idiomas, entre ellos, inglés y español.

La herramienta Freeling es utilizada para: *a)* separar la consulta del usuario en tokens, *b)* etiquetar gramaticalmente cada token de acuerdo a su función dentro de la consulta (Part of Speech) y *c)* obtener los lemas de cada token. Por ejemplo, si el usuario introduce la consulta “*Show me Internet sales amount as per customer.*” en el idioma inglés, ésta es separada en tokens como se muestra a continuación:

“*Show*” | “*me*” | “*Internet*” | “*sales*” | “*amount*” | “*as*” | “*per*” | “*customer*” | “.”

En seguida, se etiqueta gramaticalmente cada uno de los tokens y se determinan sus lemas. En la Tabla 1 se muestra el resultado del procesamiento realizado por Freeling. La información gramatical es asignada a sus respectivos tokens en la memoria del MI para ser enviada a las siguientes fases de procesamiento. Cada una de las siguientes fases actualizará la información de los tokens.

**Tabla 2.** Ejemplo de mapeos asignados a tokens.

TOKEN	LEMA	POS	SIGNIFICADO POS
Show	Show	VB	
me	me	PRP	
Internet	internet	NP	[Internet Sales Order Details].[Sales Order Line].[Sales Order Line] [Measures].[Internet Average Sales Amount] [Measures].[Internet Sales Amount] [Measures].[Internet Tax Amount] [Measures].[Internet Total Product Cost]
Sales	sales	NNS	[Employee].[Sales Person Flag].[Sales Person Flag] [Internet Sales Order Details].[Sales Order Line].[Sales Order Line] [Measures].[Internet Average Sales Amount] [Measures].[Internet Sales Amount] [Measures].[Reseller Average Sales Amount] [Sales Channel].[Sales Channel].[Sales Channel] [Sales Reason].[Sales Reason].[Sales Reason] [Sales Summary Order Details].[Sales Orders].[Sales Orders]
amount	amount	NN	[Measures].[Reseller Average Sales Amount] [Measures].[Internet Average Sales Amount] [Measures].[Internet Extended Amount] [Measures].[Internet Sales Amount] [Measures].[Internet Tax Amount] [Measures].[Reseller Sales Amount] [Measures].[Reseller Tax Amount] [Measures].[Sales Amount Quota]
as	as	IN	
per	per	IN	
customer	customer	NN	[Customer].[Customer].[Customer] [Clustered Customers].[Customer Clusters].[Customer Clusters] [Customer].[City].[City] [Measures].[Customer Count] [Measures].[Growth in Customer Base]
.	.	FP	

### 3.4. Procesamiento semántico

El objetivo de esta fase es identificar sin ambigüedad conceptos clave en la consulta del usuario. Un concepto clave es aquel token o conjunto de tokens (token combinado) que mapean a elementos de la ontología.

Para identificar los conceptos clave se recorre cada uno de los tokens etiquetados gramaticalmente como sustantivo, adjetivo o verbo. Por cada token, el MI genera una serie de consultas SPARQL [7] con el fin de acceder al conocimiento modelado en la

**Tabla 3.** Ejemplo de un token compuesto.

Token	Lema	PoS	Significado PoS
Show	Show	VB	
me	me	PRP	
Internet sales amount	Internet sales amount	NP	[Measures].[Internet Average Sales Amount] [Measures].[Internet Sales Amount]
as	as	IN	
per	per	IN	
customer	customer	NN	[Customer].[Customer].[Customer] [Customer].[City].[City] [Measures].[Customer Count] [Measures].[Growth in Customer Base]
.	.	FP	

ontología para identificar si el token coincide textualmente con algún individuo de la clase *CToken*.

Si existe alguna relación token – *CToken*, el MI genera nuevamente una serie de consultas SPARQL para identificar, a través de la propiedad de objeto *mappingObject*, los elementos de la ontología (*CCube*, *CDimension*, *CHierarchy* y/o *CMeasure*) mapeados por el individuo *CToken*. Todos los elementos mapeados son asignados en memoria al token correspondiente. En la Tabla 2 se presentan sólo algunos ejemplos de mapeos asignados a tokens, ya que la base de datos AdventureWorksDW2014 almacena varios cubos multidimensionales compuestos por varias dimensiones, jerarquías y métricas.

Posteriormente, el MI busca reducir el número de tokens de la consulta. Para este fin, se analizan los mapeos asignados a cada token y se aplican algunas reglas gramaticales sencillas, definidas a priori. Los tokens que tienen asignados al menos un mapeo en común y que satisfacen las reglas gramaticales son combinados para formar un solo token compuesto, conservando los mapeos en común y eliminando los mapeos diferentes. Al reducir el número de tokens y al reducir sus mapeos, se simplifica el proceso de generación de la consulta MDX. Por ejemplo, las reglas gramaticales indican al MI que puede intentar combinar los tokens *Internet*, *sales* y *amount*, pero no *customer*. A continuación, el MI analiza los mapeos de estos tokens y determina que tienen en común los mapeos: [Measures].[Internet Average Sales Amount] y [Measures].[Internet Sales Amount]. Por tal motivo, el MI los combina en uno solo token y descarta los mapeos restantes.

Adicionalmente, el MI determina que ambos mapeos pertenecen al mismo cubo, *Adventure\_Works*. Con respecto a *customer*, los mapeos que no pertenecen al cubo *Adventure\_Works* son descartados. En la Tabla 3 se puede observar que *Internet sales amount* ahora es un token compuesto, el cual contiene solamente los mapeos [Measures].[Internet Average Sales Amount] y [Measures].[Internet Sales Amount]. Adicionalmente, también se puede notar que el mapeo [Clustered Customers].[Customer Clusters].[Customer Clusters] de *customer* fue descartado debido a que pertenece al cubo *Mined\_Customers*.

Por último, el MI identifica los tokens ambiguos con el propósito de mostrar un diálogo de desambiguación que permita al usuario especificar el contexto. Un token que poseen mapeos a diferentes elementos, es decir que mapean a diferentes dimensiones, jerarquías o métricas es considerado como ambiguo.

### **3.5. Generación de consulta MDX**

Después de que el MI logró identificar sin ambigüedad los tokens y sus respectivos mapeos en los procesos descritos en las secciones anteriores, se procede a generar la consulta MDX.

De manera general, para generar una consulta MDX, se analiza cada uno de los tokens que poseen mapeos asignados. Es importante mencionar que, para generar la consulta, se verifica que todos los mapeados de los tokens pertenezcan a un mismo cubo. Esto es debido a que, hasta nuestro conocimiento, no se puede acceder a más de un cubo en una sola consulta MDX.

Todos los elementos que mapean a individuos de la clase *CMeasure*, son agregados a la cláusula SELECT. Posteriormente, se incluye la cláusula ON COLUMNS.

Todos los elementos que mapean a individuos que no pertenecen a la clase *CMeasure*, son agregados a la cláusula SELECT. Posteriormente se incluye la cláusula ON ROWS. Es importante mencionar que, si estos elementos son de diferentes dimensiones del cubo, se incluye la cláusula CROSSJOIN.

La consulta MDX generada a partir de la consulta utilizada como ejemplo en la Sección 3.2 se presenta a continuación:

```
SELECT {[Measures].[Internet Sales Amount]} on Columns,  
        {[Customer].[Customer].[Customer]} on Rows  
FROM [Adventure Works]
```

## **4. Discusiones**

A pesar de que el desarrollo de ILNs tuvo sus inicios en los años 60, actualmente no son capaces de responder correctamente todas las preguntas formuladas por los usuarios y, en consecuencia, existen muchos problemas que aún continúan abiertos a la investigación.

Sin lugar a duda, el lenguaje natural representa por sí mismo uno de los principales problemas a vencer debido a su riqueza lingüística y a diversas situaciones que en este se presentan, tales como ambigüedades (léxicas, sintácticas, semánticas, etc.), elipsis (nominal, verbal, preposición, etc.), anáforas (nominal, pronominal, etc.), entre otros. Es del principal interés de la comunidad científica hacer frente a estas situaciones, ya que limitan la capacidad de entendimiento de una interfaz, afectando negativamente su desempeño. Adicionalmente, entre los problemas inherentes a una interfaz se encuentran los relacionados con el proceso de traducción, con la portabilidad a diferentes bases de datos y con la forma de representar el conocimiento, entre otros. Abordar satisfactoriamente la mayoría de estos problemas depende del conocimiento del que disponga la interfaz.

Con el desarrollo de este trabajo se intenta explorar qué elementos deben conformar el conocimiento de una interfaz, cómo obtenerlos de una manera automatizada para

facilitar la portabilidad de la interfaz y cómo modelarlos. Con este trabajo se pretende construir una base sólida que permita abordar problemas complejos relacionados con el lenguaje natural, como los mencionados anteriormente.

Hasta el momento, el conocimiento generado por el MGC descrito en la Sección 3.1 ha servido para probar el funcionamiento del MI descrito en la Sección 3.2. En base al conocimiento generado, el MI ha logrado traducir consultas sencillas. No obstante, se ha identificado que para mejorar el desempeño del MI es necesario mejorar la representación semántica mostrada en la Fig. 3 y mejorar el procesamiento semántico que el MI realiza.

## **5. Conclusiones y trabajos futuros**

En la Sección 5.1 se presentan las conclusiones obtenidas a partir del desarrollo de este proyecto y en la Sección 5.2 se presentan los trabajos futuros a realizar a corto plazo.

### **5.1. Conclusiones**

Las ILN son excelentes herramientas para que usuarios inexpertos accedan a información almacenada en bases de datos, ya que no requieren que éstos tengan conocimientos sobre bases de datos, ni necesitan aprender un lenguaje en especial. Su facilidad de uso las hace idóneas para esta labor.

A pesar de que su desarrollo inició hace más de cuatro décadas, aún no han logrado alcanzar el desempeño esperado por los usuarios.

Una pieza clave para lograr un buen desempeño es el conocimiento del dominio, así como el lingüístico, que una ILN debe poseer. Adicionalmente, el modelado de los elementos que componen el conocimiento y el mecanismo de representación de conocimiento juegan un papel importante.

### **5.2. Trabajos futuros**

En primera instancia se trabajará en mejorar la representación semántica mostrada en la Fig. 3 y mejorar el procesamiento semántico que realiza el MI para mejorar su desempeño.

Otro aspecto a abordar a corto plazo es integrar el uso de rebanadores en el proceso de la generación de la consulta MDX. Los rebanadores se deben agregar a la cláusula WHERE. Los rebanadores pueden ser elementos del mismo tipo que los elementos que constituyen la cláusula SELECT. Por tal motivo, integrar el uso de rebanadores implica mejorar el procesamiento de lenguaje natural que el MI realiza, ya que se debe determinar si un elemento identificado se debe agregar a la cláusula SELECT o a la cláusula WHERE.

Posteriormente, se evaluará el desempeño del MI migrando algunas bases de datos utilizadas en la literatura de ILNs a cubos multidimensionales.

Esto debido a que algunas de estas bases de datos están disponibles en la literatura y disponen de un corpus de consultas para realizar la evaluación. No obstante, se analizará la creación de un corpus de consultas para la base de datos multidimensional



AdventureWorksDW2014, ya que posee un número de tablas considerable que implica un reto para cualquier ILN. Las métricas que se utilizarán para evaluar el desempeño serán las métricas *Precision* y *Recall*, comúnmente utilizadas en la literatura para la evaluación de este tipo de interfaces.

Finalmente, a mediano plazo se diseñará un módulo que permita a los usuarios gestionar el conocimiento del MI con el fin de mejorar su funcionamiento y asegurar su utilidad dentro del área de Inteligencia de Negocios.

**Agradecimientos.** Agradecemos a PRODEP por el apoyo brindado al proyecto titulado *Conocimiento Semántico en una Interfaz de Lenguaje Natural Portable para Acceder a Información de Bases de Datos Multidimensionales en el Área de Negocios Inteligentes*, con el cual este artículo fue posible. UACJ-PTC-373.

## Referencias

1. Kuchmann-Beauger, N., Aufaure, M.: A Natural Language Interface for Data Warehouse Question Answering. In: Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems, pp. 201–208 (2012)
2. Saias, J., Quaresma, P., Salgueiro, P., Santos, T.: BINLI: An Ontology-Based Natural Language Interface for Multidimensional Data Analysis. In: Intelligent Information Management, pp. 225–230 (2012)
3. Prat, N., Akoka J., Wattiau, I.: Transforming Multidimensional Models into OWL-DL Ontologies. In: Proceedings - International Conference on Research Challenges in Information Science, pp. 1–12 (2012)
4. Popescu, A., Armanasu, A., Etzioni, O., Ko, D., Yates, A.: Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. In: Proceedings of the 20th international conference on Computational Linguistics, pp. 141 (2004)
5. Stanford Log-linear Part-Of-Speech Tagger, <https://nlp.stanford.edu/software/tagger.shtml> (2018)
6. Freeling Features, <http://nlp.lsi.upc.edu/freeling/node/4> (2018)
7. SPARQL Query Language for RDF, <https://www.w3.org/TR/rdf-sparql-query/> (2018)



## Minería de opiniones aplicada a la evaluación docente de la Universidad Autónoma de Ciudad Juárez

Rafael Jiménez Castro, Vicente García, Rogelio Florencia Juárez,  
Gilberto Rivera Zarate, Francisco López Orozco

Universidad Autónoma de Ciudad Juárez,  
División Multidisciplinaria en Ciudad Universitaria,  
México

al117955@alumnos.uacj.mx  
{vicente.jimenez, rogelio.florencia, gilberto.rivera, francisco.orozco}@uacj.mx

**Resumen.** La Universidad Autónoma de Ciudad Juárez lleva a cabo una evaluación docente cada semestre con la finalidad de encontrar las fortalezas, debilidades y áreas de oportunidad entre los profesores. En este artículo mostramos como la minería de opiniones puede ser útil para etiquetar comentarios de estudiantes en positivos y negativos. Para ello, se construyó una base de datos reales construida a partir opiniones obtenidas de cinco profesores de la UACJ a lo largo de cuatro años, abarcando un total de 20 materias. Sobre la base de datos se utilizaron técnicas de procesamiento de lenguaje natural para normalizar los datos contenidos en ella. Resultados experimentales utilizando los clasificadores 1-NN y SMO mostraron que es posible etiquetar de forma automática comentarios en positivos y negativos con una exactitud del 79.46%.

**Palabras clave:** minería de opiniones, evaluación docente, estudiantes, clasificación.

### Opinion Mining for Instructor Evaluations at the Autonomous University of Ciudad Juarez

**Abstract.** The Universidad Autónoma de Ciudad Juárez performs an instructor evaluation each semester to find strengths, weaknesses, and areas of opportunity during the teaching process. In this article we show how opinion mining can be useful for labeling student comments as positives and negatives. For this, a database was created using real opinions obtained from 5 professors of the UACJ over 4 years, covering a total of 20 subjects. Natural language processing techniques were used on the database to normalize its data. Experimental results using 1-NN and SMO classifiers shows that it is possible to automatically label positive and negative comments with an accuracy of 79.46%.

**Keywords:** opinion mining, instructor evaluation, students, classification.

## 1. Introducción

Las opiniones son actividades centrales para casi todos los seres humanos y cuando tenemos que tomar decisiones es importante saber la valoración de los demás, por ello, las opiniones son una fuente valiosa de información. El análisis de sentimientos o minería de opiniones es un área que clasifica de forma automática los sentimientos, expresados por una persona sobre un objeto determinado, en positivos, negativos o neutros [1]. La minería de opiniones es utilizada por las compañías para conocer las percepciones que tienen los clientes sobre sus productos o servicios, así como para identificar áreas de oportunidad o mejorar las estrategias de mercadotecnia utilizadas [2].

La evaluación docente que realiza la Universidad Autónoma de Ciudad Juárez (UACJ) cada semestre es un método por el cual, se registra la opinión escrita de los estudiantes para identificar las fortalezas, debilidades y áreas de oportunidad en el desempeño de los profesores [3].

Durante el periodo de evaluación, la jefatura de evaluación docente pone a disposición de los alumnos una plataforma en la que, además de otras métricas, aparecen dos casillas para que puedan escribir de forma libre comentarios positivos y negativos sobre sus maestros durante el semestre en curso. En este proceso, el estudiante erróneamente escribe comentarios negativos en la casilla de positivos y viceversa, así como una combinación de ambos. Esto ocasiona que la evaluación docente no dé una fácil retroalimentación al docente al estar todos los comentarios positivos y negativos mezclados.

En este artículo se emplean técnicas de análisis de sentimientos a las opiniones emitidas por los estudiantes, con la finalidad de categorizar los comentarios en positivos y negativos. Para ello, se construyó una base de datos con opiniones de estudiantes emitidas durante cuatro años, las cuales fueron categorizadas manualmente en positivas y negativas, para posteriormente construir vectores de características, los cuales fueron empleados en el entrenamiento de una máquina de soporte vectorial (SMO) y el algoritmo de los k-vecinos más cercanos (1-NN). Un trabajo en la misma línea fue presentado por Gutiérrez et al. [4], en el cual se analiza el desempeño de los maestros utilizando reseñas que hacían por tuits los alumnos de la Universidad Politécnica de Aguascalientes.

El artículo está organizado de la siguiente manera. La Sección 2 se describe brevemente los trabajos relacionados en minería de opiniones. En la Sección 3, la metodología empleada en el desarrollo de la investigación. En la Sección 4 se muestra la configuración experimental adoptada. Posteriormente, en la Sección 5 se describen y discuten los resultados. Finalmente, en la Sección 6 se concluye y proponen líneas futuras de investigación.

## 2. Trabajos relacionados

La minería de opiniones tiene un gran campo de aplicación. Existen compañías grandes y pequeñas que sólo se dedican a la minería de opiniones [5,6,7,8]; otras, la utilizan como una parte fundamental de sus operaciones [9,10,11].

## 2.1. Reseñas en la web

La minería de opiniones en la web [12] se emplea para automatizar el mantenimiento de reseñas y opiniones, ya que las redes sociales son una fuente rica de información a gran escala. Los usuarios las utilizan para expresar sus sentimientos sobre varios temas, muchos de ellos, sobre productos. Estos comentarios se pueden clasificar en positivos, negativos o neutros y de ellos se puede extraer información valiosa para los reportes de mercado de las compañías. Algunos ejemplos de aplicaciones en el mercado son:

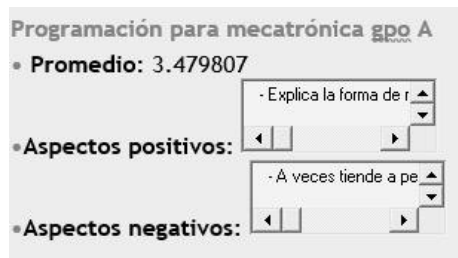
- *Meaning cloud* [13]: Es una aplicación en línea que ofrece los servicios de clasificación de textos, extracción de temas, análisis de sentimiento, identificación de idioma, tematización y análisis morfosintáctico, reputación corporativa y *clustering* de texto. La aplicación tiene soporte para varios lenguajes, entre ellos el inglés, español, francés, italiano, portugués y catalán. La forma en la que funciona Meaning Cloud es por medio un *plugin* de Excel o por medio de un API a su servicio en línea que contiene todas las librerías en un solo paquete para realizar la extracción de opiniones.
- Vivek Sentiment [14]: Esta aplicación web fue creada utilizando un modelo naïve Bayes y utiliza reseñas de la base de datos de películas IMDB.com. El sistema evalúa los comentarios como positivos, negativos o neutros, además de dar el nivel de confianza en el resultado. La aplicación tiene un API para poder utilizar el clasificador de forma externa, pero solo si se cuenta con conexión a Internet. Otra de las limitantes que tiene es el lenguaje, ya que su dominio abarca solamente el idioma inglés.

## 2.2. En negocios

Las compañías interesadas en conocer las percepciones de los clientes sobre sus productos o servicios emplean minería de opiniones. La información recabada de encuestas es clasificada, lo que permite mejorar un producto, identificar áreas de oportunidad o mejorar las estrategias de mercadotecnia utilizadas [12]. Un ejemplo de ello es Meltwater [15], la cual es una consultora que ofrece servicios y paquetes que ayudan a crear un análisis sobre la presencia de la marca ante los consumidores y la competencia. Entre los análisis que se realizan, está el análisis de sentimientos, el cual monitorea qué tan bien recibido fue un mensaje de la compañía en redes sociales.

## 2.3. Inteligencia gubernamental

La minería de opiniones gubernamental extrae el comportamiento y opiniones públicas sobre cuestiones políticas. El uso del análisis de sentimientos ayuda a identificar las opiniones de los votantes con relación a un candidato antes de las elecciones para así mejorar las estrategias de campaña [12]. Un ejemplo de ello es el Instituto Nacional de Estadística, Geografía e Informática (INEGI), el cual desarrolló una herramienta que clasificó 63 millones de tuits con georreferencia escritos entre el



**Fig. 1.** Forma en la cual los comentarios de la evaluación docente aparecen a los profesores.

2014 y 2015. Esto con la finalidad de generar estadísticas de movilidad y turismo, así como conocer el estado de ánimo de la población por estado del país [16].

### 3. Metodología

Los pasos que se llevaron a cabo para la clasificación de comentarios de la evaluación docente fueron los siguientes: 1) Recolección de comentarios de la evaluación docente, 2) Preprocesamiento de los comentarios, 3) Creación de un vector de características y 4) Clasificación de los comentarios. Estos pasos se describirán brevemente en las siguientes secciones.

#### 3.1. Recolección de comentarios

La recolección de comentarios se llevó a cabo durante el mes de enero del 2017 con ayuda de los profesores de la División Multidisciplinaria de la UACJ en Ciudad Universitaria (CU). Los comentarios corresponden a 4 años, 20 materias y 5 profesores. Para la extracción de los comentarios, cada profesor realizó los siguientes pasos:

1. Ingreso a su página de resultados en la evaluación docente.
2. Selección de una materia ya impartió con anterioridad.
3. Localizar el área de comentarios, como la que se muestra en la Fig. 1.

Al tener en pantalla las casillas de comentarios se escogieron todos los positivos y negativos, los cuales se almacenaron en un archivo de texto simple por separado.

#### 3.2. Preprocesamiento de los comentarios

Antes de crear el vector de características de los comentarios, se realizó un preprocesamiento manual, el cual consistió en la validación de comentarios. Durante esta etapa se eliminaron los comentarios que no tuvieran relevancia a la evaluación docente, así como redistribuir los comentarios positivos que estaban en el archivo de comentarios negativos, así como también los negativos que estaban en el archivo de comentarios positivos. Algunos ejemplos de comentarios no relevantes son:

- “Profesor en serio yo nunca le explique a Yair el método mini, Max el asumió que sí, pero está tonto. Se lo juro.”
- Validación de escritura: La validación de escritura fue la etapa en la cual se corrigió la escritura en los comentarios, editando las palabras que estuvieran mal escritas. Ejemplo: “inpuntual” → “impuntual”
- Eliminación de caracteres y agregación de punto final: Ya que FreeLing requiere que todos los comentarios terminen con un punto final, este punto se agregó mientras se hacía la eliminación de otros caracteres en el comentario como el carácter de guión (“ - “) que aparece al principio de todos los comentarios recolectados.

También se eliminaron otros símbolos como guiones, emojis e ideogramas. Asimismo, se agregó punto al final de una frase. Ejemplo de ello es:

- “- Tiene dominio sobre su tema” → “Tiene dominio sobre su tema.”  
“es una muy buena maestra ❤️” → “es una muy buena maestra”

También se utilizó preprocesamiento automático con la función de normalización, la cual consistió en:

- Cambio de mayúsculas a minúsculas para eliminar posibles errores al comparar las palabras.
- Eliminación de acentos en las palabras ya que los alumnos no acentúan las palabras y ocasionaba que las palabras no fueran detectadas.
- Eliminación de caracteres, comillas y guión bajo. Este tipo de caracteres no contribuyen nada.

### **3.3. Creación de un vector de características**

Un vector de características es un vector de n-elementos o atributos que describen un objeto. Para poder convertir un comentario a un vector, se definió previamente los atributos que lo conformaran. A partir de una lista de palabras indicadoras de opinión en español independiente del dominio, que contienen palabras positivas y negativas se construyó el vector de características. La lista fue proporcionada por la Red Temática en Tratamiento de la Información Multilingüe y Multimodal [17], la cual consta de 2509 palabras positivas y 5626 palabras negativas.

Para reducir la dimensionalidad del vector, se utilizó FreeLing [18]. Esta librería permite extraer el lema de cada una de las palabras y excluir lemas duplicados, lo que produjo una disminución a 1313 palabras positivas y 2949 negativas.

Para disminuir aún más la dimensionalidad, se realizó un análisis de la raíz de las palabras. Para ello se tomó una palabra y se buscó entre todas las palabras del mismo archivo (positivo o negativo) aquellas que tengan la misma raíz. Esto disminuyó el número de características a solo 584 palabras positivas y 1270 palabras negativas.

**Tabla 1.** Ejemplo de palabras positivas y negativas como características de un vector.

Lista de palabras	Bueno	proactivo	decisivo	mejor
	impuntual	fastidio	aburrir	inexperto

**Tabla 2.** Vector resultante del comentario con el vector de características.

Lista de palabras	Bueno	proactivo	decisivo	mejor
<b>Vector</b>	1	0	0	1
Lista de palabras	impuntual	fastidio	aburrir	inexperto
<b>Vector</b>	1	0	0	0

**Tabla 3.** Descripción breve de la base de datos utilizada en los experimentos.

Numero de instancias			Numero de atributos		
Positivos	Negativos	Total	Positivos	Negativos	Total
187	110	297	591	1225	1816

La creación de un vector de características se llevó a cabo a partir de las palabras raíz de tipo positivo y negativo como se muestra en la Tabla 1, donde las palabras bueno, proactivo, decisivo y mejor son ejemplos de palabras positivas e impuntual, fastidio, aburrir e inexperto son palabras negativas, es decir, la dimensionalidad del vector está determinada por las palabras raíces de tipo positivo o negativo.

Por ejemplo, dado un comentario “Es un buen profesor, pero podría ser mejor si no fuera tan impuntual”, este se preprocesa con la función de normalización antes de ser convertido a tokens con ayuda de FreeLing, lo cual quedaría como {“es”, “un”, “buen”, “profesor”, “pero”, “podría”, “ser”, “mejor”, “si”, “no”, “fuera”, “tan”, “impuntual”}, luego se extrae el lema de las palabras {“ser”, “uno”, “bueno”, “profesor”, “pero”, “podría”, “ser”, “mejor”, “si”, “no”, “ser”, “tan”, “impuntual”}. Posteriormente, se indica en cada atributo si la palabra existe con un valor de 1 y en el caso contrario con un 0. Un ejemplo de ello se puede observar en la Tabla 2.

#### 4. Configuración de experimentos

Todas las pruebas fueron realizadas con el software WEKA, empleando para ello una validación cruzada de 5 particiones y la base de datos que se detalla brevemente en la Tabla 3. Los comentarios contenidos en la base de datos pasaron por un preprocesamiento que se describió en la sección 3.2., donde finalmente se pudieron construir 187 y 110 comentarios positivos y negativos, respectivamente.

La dimensionalidad del vector es 1816, de los cuales 591 son atributos que describen palabras positivas y 1225 palabras negativas.

Para los clasificadores SMO (Sequential Minimal Optimization) y 1-NN se utilizaron las configuraciones por defecto en WEKA donde el clasificador SMO utiliza un kernel polinomial, C igual a 1 e información normalizada y el clasificador 1-NN utiliza un kernel de distancia euclidiana y un K=1.



**Tabla 4.** Matriz de confusión de dos clases.

		Etiqueta Predicha	
		Positiva	Negativa
Etiqueta Real	Positiva	D	c
	Negativa	B	a

**Tabla 5.** Tabla de resultados de exactitud global.

	SMO	1-NN
Precisión Global	79.46%	76.76%

Como métricas de evaluación para medir el rendimiento de los clasificadores SMO y 1-NN, se utilizó una matriz de confusión de dos clases como se muestra en la Tabla 4. Con esta matriz podremos determinar el rendimiento de un clasificador con las fórmulas subsecuentes.

La exactitud se determinó con la fórmula:

$$exactitud = \frac{a + d}{a + b + c + d}. \quad (1)$$

La tasa de positivos verdaderos (TP) es la proporción de instancias positivas que fueron correctamente clasificadas, y se determina con la fórmula:

$$TP = \frac{d}{c + d}. \quad (2)$$

La tasa de negativos verdaderos (TN) es la proporción de instancias negativas que fueron correctamente clasificadas, y se determina con la fórmula:

$$TN = \frac{a}{a + b}. \quad (3)$$

La tasa de falsos positivos (FP) es la proporción de instancias negativas clasificadas como positivas, se determina con la fórmula:

$$FP = \frac{b}{a + b}. \quad (4)$$

La tasa de falsos negativos (FN) es la proporción de instancias negativas clasificadas como positivas, se determina con la fórmula:

$$FN = \frac{c}{c + d}. \quad (5)$$

## 5. Resultados y discusiones

Los resultados globales de exactitud se muestran en la Tabla 5, donde podemos observar que el algoritmo SMO obtuvo los mejores resultados.

**Tabla 6.** Tabla de precisión por clase.

	<b>SMO</b>	<b>1-NN</b>
<b>Positivos</b>	91.44%	89.80%
<b>Negativos</b>	59.10%	54.54%

**Tabla 7.** Matriz de confusión para el clasificador SMO.

	<b>SMO</b>	
	<b>Positivo</b>	<b>Negativo</b>
<b>Positivos</b>	171	16
<b>Negativos</b>	45	65

**Tabla 8.** Matriz de confusión para el clasificador 1-NN.

	<b>1-NN</b>	
	<b>Positivo</b>	<b>Negativo</b>
<b>Positivos</b>	168	19
<b>Negativos</b>	50	60

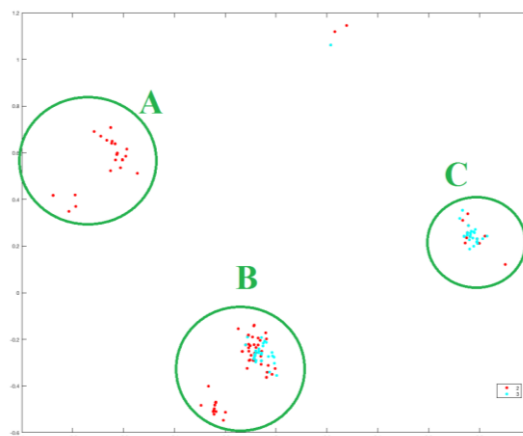
En la Tabla 6 se muestra la precisión por clase para cada uno de los clasificadores. Como se puede observar los comentarios negativos obtienen una baja tasa de clasificación entre el 54% y el 60%. Mientras que los comentarios positivos obtienen resultados mayores al 88%. Los bajos resultados negativos, afectan a la tasa global de clasificación.

Con la finalidad de analizar el comportamiento por clase de cada uno de los clasificadores, la Tabla 7 y 8 muestran la matriz de confusión promedio para los clasificadores SMO y 1-NN, respectivamente.

La matriz de confusión M muestra que el modelo clasificó correctamente los elementos de la parte superior izquierda  $M[1,1]$  y la parte inferior derecha  $M[2,2]$ . En la parte inferior izquierda  $M[2,1]$  y superior derecha  $M[1,2]$ , se muestran los falsos positivos y negativos, respectivamente.

De la matriz de confusión para el clasificador SMO en la Tabla 7, se obtiene un TP de 0.9144, TN de 0.5910, el FP de 0.4090 y el FN de 0.0855. Dadas las tasas de FP y FN, podemos observar que este algoritmo de clasificación tiene un menor índice de FP que de FN, lo cual indica que hubo un porcentaje menor de comentarios negativos clasificados como positivos que de comentarios positivos clasificados como negativos. Para el clasificador 1-NN en la Tabla 8, se obtiene un TP de 0.8983, un TN de 0.5454, el FP de 0.4545 y el FN de 0.1016.

Este algoritmo a diferencia del algoritmo SMO, tiene una tasa de FP más alta que FN, por lo que más comentarios negativos fueron clasificados como positivos que viceversa.



**Fig. 2.** Trazado en dos dimensiones de las instancias utilizadas para el entrenamiento.

Dadas los FP y FN de los dos clasificadores, el algoritmo SMO tiene el menor índice de falsos positivos y falsos negativos. Lo que nos indica que con este algoritmo se tienen menos comentarios positivos clasificados como negativos y menos comentarios negativos clasificados como positivos.

Para darnos una mejor idea de cómo están ubicadas las instancias en el espacio y del porqué de los resultados, se utilizó el escalamiento multidimensional (MDS). Este último se suele emplear para visualizar datos con una alta dimensionalidad, en bajas dimensiones. En la Fig. 2 podemos ver cómo están localizadas las instancias positivas (rojo) y negativas (cian) en un plano cartesiano. En la figura se pueden observar tres grupos de datos marcados con las letras A, B y C. En el punto A es un agrupamiento de datos positivos que no presenta ni dentro ni cerca de dicho grupo otro tipo dato. Por lo que se podría decir que las instancias que pertenecen a este espacio serán correctamente clasificadas.

Lo contrario ocurre en los puntos B y C, en donde, existe un solapamiento entre muestras positivas y negativas. Es precisamente en estas zonas donde el clasificador suele cometer los errores. Mirando con detalle el punto C, se podría decir que los comentarios positivos en dicha zona serán clasificados como negativos. Mientras que en la zona B, la mayoría de los comentarios negativos serán clasificados como positivos.

Este solapamiento en el punto B se debe a que los comentarios pueden contener las mismas palabras tanto en comentarios positivos, como en comentarios negativos, por ejemplo, la palabra “bien”, esta palabra se repite tanto en comentarios negativos como:

- “realmente la puedo considero como una maestra que enseña del todo bien aunque cumpla con todos los puntos previos, considero que no es su culpa, el salón que nos tocó este semestre se batalla mucho para escuchar (d4-210).”

También ocurre en comentarios positivos como:

- “sus grupos casi siempre están vacíos, eso habla bien de él, ya que da la clase bien y les da a los alumnos la calificación que se merecen, no deberían abrir esta materia en veranos, los que no hacen nada buscan como pasar y si se gradúan eso da mala imagen a la uacj en la industria, a veces se vuelve difícil conseguir un buen trabajo cuando tienen catalogado a los alumnos de la uacj como inútiles y flojos.”

Al tener este tipo de comentarios que contienen la misma palabra y que son subjetivamente ambiguos por una cantidad similar de palabras positivas y negativas que contienen, se podría considerar este comentario como neutro.

## 6. Conclusiones

En este trabajo se empleó la minería de opiniones a una base de datos de comentarios de estudiantes provenientes de la evaluación docente de la UACJ. Los comentarios corresponden a cuatro años, 20 materias y cinco profesores. Para ello, se realizaron una serie de pasos que fueron desde la captura de los comentarios, pasando por el procesamiento hasta el entrenamiento y prueba con dos clasificadores.

Los resultados mostraron que de los clasificadores empleados, SMO y el 1-NN, el mejor fue el primero con el cual se pudo alcanzar una exactitud de clasificación de un 79.46. Asimismo, se pudo observar que los errores fueron producidos por una serie de comentarios que fueron etiquetados como positivos o negativos, pero que se caracterizan por ser comentarios que pudieran ser considerados neutros.

Como trabajos futuros se probará integrar la librería CoreNLP de la Universidad de Stanford para ayudar con las dependencias entre palabras y la identificación de entidades como personas, lugar y organización. Asimismo, se agregará una nueva etiqueta de clase de comentario denominado neutro. Finalmente, se propone emplear técnicas de preprocesamiento como edición, condensado, entre otros.

**Agradecimientos.** Este trabajo fue parcialmente financiado por el Programa Para el Desarrollo Profesional Docente, para el tipo superior (PRODEP), con clave UACJ-PTC-373.

## Referencias

1. Cortizo, J.C.: Minería de Opiniones. BrainSINS, [www.brainsins.com/es/blog/mineria-opiniones/3555](http://www.brainsins.com/es/blog/mineria-opiniones/3555) (2016)
2. Huddy, G.: What Is Sentiment Analysis? The importance of understanding how your audience feels about your brand. Crimson Hexagon, <https://www.crimsonhexagon.com/blog/what-is-sentiment-analysis/> (2018)
3. Universidad Autónoma de Ciudad Juárez.: Evaluación Docente. <http://www.uacj.mx/sa/ed/Paginas/default.aspx> (2018)
4. Gutiérrez, G., Ponce, J., Ochoa, A., Álvarez, M.: Analyzing Students Reviews of Teacher Performance Using Support Vector Machines by a Proposed Model. Communications in Computer and Information Science, 820, pp. 113–122 (2018)

5. Revuze: Revuze Products. Revuze. <http://revuze.it/product/> (2018)
6. Aspectiva.: Aspectiva About Us. Aspectiva, <http://www.aspectiva.com/company/> (2018)
7. Brandwatch.: How it works. Brandwatch, <https://www.brandwatch.com/>. (2018)
8. Google: Sentiment Analysis Tutorial. Google Analytics. <https://cloud.google.com/natural-language/docs/sentiment-tutorial> (2018)
9. Amazon Web Services: Big Data on AWS. Amazon.com <https://aws.amazon.com/big-data/> (2018)
10. Waterloo University: Capital One Data Mining Cup. Waterloo University. <https://uwaterloo.ca/computing-financial-management/events/capital-one-data-mining-cup> (2014)
11. Jadhavar, R., Komaraji, A.K.: Sentiment Analysis of Netflix and Competitor Tweets to Classify Customer Opinions. SAS Global Forum (2018)
12. Smeureanu, I., Bucur, C.: Applying Supervised Opinion Mining Techniques on Online User Reviews. Informatica Economică, 16(2), pp. 81–91 (2012)
13. Meaning Cloud: General questions. Meaning Cloud. <https://www.meaningcloud.com/> (2018)
14. Narayanan, V., Arora, I., Bhatia, A.: Fast and accurate sentiment classification using an enhanced Naive Bayes model Cornell. <http://sentiment.vivekn.com/about/> (2016)
15. Meltwater: About Meltwater. Meltwater: <https://www.meltwater.com/about/> (2016)
16. Instituto Nacional de Estadística y Geografía.: Estado de ánimo de los tuiteros en los Estados Unidos Mexicanos. [http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod\\_serv/contenidos/espanol/bvinegi/productos/nueva\\_estruc/702825099718.pdf](http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod_serv/contenidos/espanol/bvinegi/productos/nueva_estruc/702825099718.pdf) (2017)
17. Molina-González, D., Martínez-Cámara, E., Martín-Valdivia, M.T., Perea-Ortega, J.: iSOL. Red Temática en Tratamiento de la Información Multilingüe y Multimodal. <http://timm.ujaen.es/recursos/isol/> (2013)
18. Freeling: FreeLing Home Page. Universitat Politècnica de Catalunya. <http://nlp.lsi.upc.edu/freeling/node/1> (2015)



## Uso de analizador de emociones en sistemas educativos inteligentes

María Lucia Barrón-Estrada<sup>1</sup>, Ramón Zatarain-Cabada<sup>1</sup>,  
Sandra Lucia Ramírez-Ávila<sup>1</sup>, Raúl Oramas-Bustillos<sup>1</sup>,  
Mario Graff Guerrero<sup>2</sup>

<sup>1</sup> Tecnológico Nacional de México, Instituto Tecnológico de Culiacán,  
México

<sup>2</sup> INFOTEC, Aguascalientes,  
México

{lbarron, rzatarain, sramirez, raul.oramas}@itculiacan.edu.mx,  
mario.graff@infotec.mx

**Resumen.** En este trabajo se presenta el desarrollo de un clasificador de frases relacionadas con el aprendizaje en el ámbito de programación de computadoras para realizar análisis de sentimientos en sistemas educativos inteligentes. El clasificador se ofrece como un servicio web que recibe un texto en español y regresa como resultado una emoción centrada en el aprendizaje. El clasificador fue entrenado con un corpus de frases en español expresadas por estudiantes al acceder a diversos objetos de aprendizaje. El corpus se creó mediante el Sistema de Evaluación de Recursos Educativos (SERE) que se encarga de recolectar frases textuales escritas en español las cuales reflejan la opinión de los estudiantes sobre los recursos educativos utilizados en el aprendizaje de diversos temas. Las opiniones (frases textuales) fueron etiquetadas para categorizarlas en diferentes emociones centradas en el aprendizaje tales como: Frustrado, Aburrido, Emocionado y Comprometido. La principal contribución de este trabajo es un analizador para el reconocimiento de emociones centradas en el aprendizaje usando frases textuales escritas en español que podrá ser utilizado por un sistema tutor inteligente para detectar emociones de los estudiantes y realizar de forma más eficiente el proceso de enseñanza con los estudiantes adaptando el contenido tanto a las necesidades cognitivas como afectivas.

**Palabras clave:** análisis de texto, análisis de sentimientos, minería de opiniones, sistemas educativos inteligentes, entorno de aprendizaje inteligente, emociones en el aprendizaje.

### Use of Emotion Analyzer in Intelligent Educational Systems

**Abstract.** In this paper, we present the development of a classifier of Spanish sentences related to learning in the field of computer programming to make sentiment analysis inside intelligent educational systems. The classifier is offered

as a web service that receives a text in Spanish and returns as a result a learning centered emotion. The classifier was trained with a corpus of Spanish phrases expressed by students after accessing different learning objects. The corpus was created through the Educational Resources Evaluation System (SERE) that is responsible for collecting textual phrases written in Spanish which reflect the students' opinion about the educational resources used in the learning of various topics. The opinions (textual phrases) were labeled to categorize them into different learning centered emotions such as: frustrated, bored, excited and engaged. The main contribution of this work is an analyzer for the recognition of learning centered emotions using textual phrases written in Spanish; which can be used by an intelligent tutor system to detect students' emotions and to execute more efficiently the teaching process with the students, and adapting the content to both cognitive and affective needs.

**Keywords:** text analysis, sentiment analysis, opinion mining, intelligent educational systems, intelligent learning environment, learning-centered emotions.

## **1. Introducción**

En un ambiente educativo tradicional los estudiantes interactúan con sus compañeros para intercambiar experiencias y colaborar en el desarrollo de proyectos o la resolución de problemas con el fin de aprender diversos temas. En el modelo constructivista, el profesor es un facilitador que proporciona herramientas y guía a los estudiantes en el proceso de construcción de su propio conocimiento. Sin embargo, los programas de estudio no son personalizados y todos los estudiantes deben acceder a los mismos temas en los mismos tiempos. Además, las emociones juegan un papel fundamental durante el aprendizaje y pueden propiciar un mejor aprovechamiento o bloquear el aprendizaje, por eso es importante detectar las emociones y actuar acorde a éstas con el fin de mantener al estudiante interesado en el tema de estudio.

Se han desarrollado diferentes tecnologías para enfrentar la necesidad de integrar las emociones en el proceso de enseñanza-aprendizaje. Entre estas tecnologías se encuentran algunos Sistemas Tutores Inteligentes (ITS, por sus siglas en inglés) y Entornos de Aprendizaje Inteligentes (ILE, por sus siglas en inglés) los cuales fueron diseñados para capturar e identificar las emociones de los usuarios, pero la mayoría de estos sistemas convencionales funcionan solo con emociones básicas [1].

En trabajos anteriores se han estudiado las emociones básicas, entre las que se encuentran: la ira, la felicidad, la tristeza y el miedo; las cuales se expresan en diferentes situaciones de la vida cotidiana [2]. Sin embargo, hay otro tipo de emociones que emergen durante las actividades de aprendizaje profundo y son llamadas emociones centradas en el aprendizaje [3,4]; en éstas un estudiante puede sentirse: frustrado, aburrido, comprometido o emocionado. Las emociones centradas en el aprendizaje juegan un papel importante en los estudiantes, ya que afectan diferentes aspectos como mecanismos cognitivos y retención de información [5,6].

Actualmente la mayoría de las herramientas para el reconocimiento de emociones en texto trabaja con corpus de frases textuales generales escritas en inglés que se recolectaron de diferentes redes sociales, aunque Tellez, Jiménez, Graff, Moctezuma, Siordia, & Villaseñor, utilizan frases en español cuyo objetivo es analizar



exhaustivamente todas las combinaciones de las transformaciones de texto para descubrir características comunes [16]. En estos trabajos, las frases no son específicas para el área de educación (aprendizaje de programación de computadoras) lo que hace necesario desarrollar un corpus de frases textuales en español que contengan información acerca de las emociones que los estudiantes experimentan durante el proceso de aprendizaje. Es por esto que se decidió crear una base de datos específica para realizar análisis de sentimientos en frases textuales en español sobre el tema de programación de computadoras.

En este trabajo, se describen los resultados generados por el clasificador de reconocimiento de emociones. Para entrenar al clasificador, se utilizó el corpus de frases textuales en español, generado por el Sistema de Evaluación de Recursos Educativos (SERE). SERE tiene como propósito mostrar a los estudiantes diversos objetos de aprendizaje (OA) solicitándole que escriba una opinión textual sobre el OA. El clasificador podrá ser utilizado para incorporar cambios y mejoras a los contenidos de los cursos y otros elementos didácticos en los ITS o ILE.

La principal contribución de este trabajo es, el desarrollo un clasificador de frases textuales en español para el reconocimiento de emociones centradas en el aprendizaje por medio de texto.

Este artículo se organiza de la siguiente forma: la sección 2 presenta los trabajos relacionados, la sección 3 expone el proceso para realizar el análisis de sentimientos. La sección 4 muestra los resultados de las pruebas del clasificador y finalmente en la sección 5 se presentan las conclusiones.

## **2. Trabajos relacionados**

En la década pasada, la proliferación de sitios de internet donde los usuarios podían expresar sus opiniones generó la necesidad de procesar automáticamente estas opiniones para obtener información relevante que pudiera ser utilizada para la toma de decisiones, surgiendo el área de Minería de Opiniones. Análisis del sentimiento o Minería de opiniones (SA u OM por sus siglas en inglés respectivamente) se define como la tarea de encontrar las opiniones de los autores sobre entidades específicas [7]. Otra definición conocida para SA es el estudio computacional de las opiniones, actitudes y emociones de las personas hacia una entidad [8].

OM extrae y analiza la opinión de la gente sobre una entidad mientras que SA identifica el sentimiento expresado en un texto y luego lo analiza. Por lo tanto, el objetivo de SA es encontrar opiniones, identificar los sentimientos que expresan y luego clasificar su polaridad [8]. Medhat et al [8], propone tres niveles principales de clasificación en SA, los cuales son:

- *Nivel de documento SA*: considera a todo el documento como una unidad básica de información y asume que el documento contiene una opinión principal expresada por su autor.
- *Nivel de oración SA*: este nivel pretende clasificar el sentimiento expresado a nivel de cada oración.
- *Nivel de aspecto SA*: pretende clasificar el sentimiento con respecto a los aspectos específicos de las entidades.

La minería de opiniones es importante porque tiene aplicaciones en diferentes áreas del conocimiento. En la educación puede ser usada para recolectar opiniones de los estudiantes sobre los temas o las estrategias de aprendizaje y esta información servirá para realizar el análisis de sentimientos y su resultado podrá ser utilizado para mejorar los materiales del curso, las estrategias de enseñanza, la personalización de contenidos, entre otros, así como también pueden ser utilizados dentro de un STI.

El análisis de sentimientos se ha utilizado en el ámbito educativo en diversos trabajos como se presenta a continuación.

Altrabsheh, Gaber, y Cocea [9] presentan SA-E, donde realizan análisis de sentimientos usando frases de Twitter para observar la retroalimentación de los estudiantes. En SA-E los estudiantes utilizaron Twitter para expresar sus opiniones sobre el material de clase, las opiniones de los estudiantes fueron utilizadas por el profesor para alterar el estilo de enseñanza de acuerdo a los resultados.

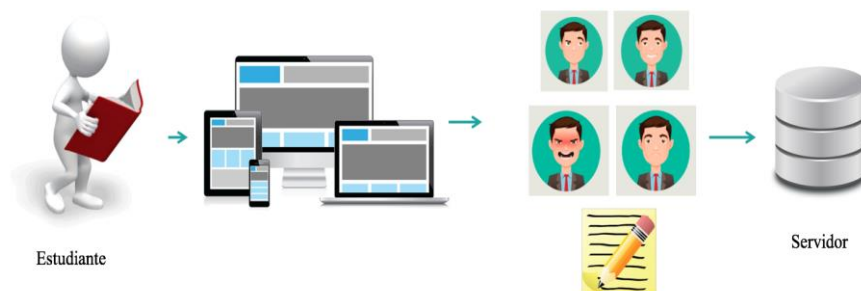
Facebook contiene un método de análisis de sentimientos que puede ser utilizado para el aprendizaje electrónico (e-learning). Los mensajes escritos por los usuarios, se usan para extraer información sobre la polaridad sentimental del mensaje (positiva, neutra o negativa), modelar la polaridad del sentimiento habitual en los usuarios y detectar cambios emocionales significativos. SentBuk [10], apoya a la detección del cambio emocional, el hallazgo emocional del amigo, la clasificación del usuario de acuerdo a sus mensajes y las estadísticas, entre otros.

El método de clasificación implementado en SentBuk sigue un enfoque híbrido, es decir, combina técnicas basadas en el léxico y la máquina. Los resultados obtenidos a través de este enfoque muestran una precisión de 83.27% en el análisis de sentimiento en Facebook. Algunas de las ventajas de e-learning es que ayuda al usuario a obtener la información sobre los sentimientos de los usuarios, de manera que esta información puede ser utilizada por sistemas adaptativos de e-learning para apoyar el aprendizaje personalizado, considerando el estado emocional del usuario y así dar una recomendación de algunas de las actividades más adecuadas para ser abordadas en el momento. Además, los sentimientos de los estudiantes hacia un curso, pueden servir como retroalimentación para los profesores ya que pueden cambiar su estrategia de enseñanza, en los Sistemas Tutores Inteligentes es posible personalizar la instrucción o mejorar el material presentado.

Por otra parte, Rowe [11] ofrece un panorama de algunos acontecimientos en el ámbito de las emociones de los estudiantes en relación con la retroalimentación, basada en de la psicología social y la educación. Altrabsheh, Cocea, and Fallahkhair [12] examinaron diferentes métodos para aprender el sentimiento de la retroalimentación de los estudiantes. Munezero, Montero, Mozgovoy, & Sutinen [13], presentan un sistema funcional para analizar y visualizar las emociones de los estudiantes, expresadas en los diarios de aprendizaje, que son instrumentos donde los estudiantes escriben sus reflexiones sobre su experiencia de aprendizaje.

### **3. Análisis de sentimientos**

En esta sección se presenta brevemente el Sistema de Evaluación de Recursos Educativos, el cual se utilizó para la creación del corpus de opiniones textuales en español para el área de programación. Además, se describe el módulo de análisis de



**Fig. 1.** Funcionamiento de Sistema de Evaluación de Recursos Educativos.

sentimientos que utiliza el corpus de opiniones para entrenamiento. Este módulo realiza la clasificación de las opiniones, esto es determinan la polaridad positiva, negativa o neutral y obtiene la emoción relacionada al aprendizaje expresada en la frase.

### 3.1. Sistema de Evaluación de Recursos Educativos (SERE)

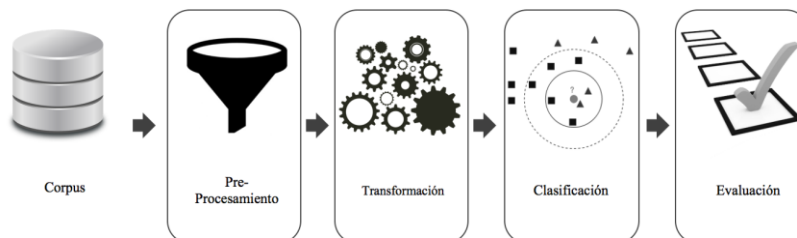
SERE fue implementado para generar un corpus de opiniones en español enfocadas en el aprendizaje. Este sistema es una aplicación Web que fue desarrollada para la plataforma .NET en Visual Studio 2013, usando la base de datos SQL Server 2008 R2 para almacenar la información de los diferentes artefactos utilizados en el ambiente educativo, como son: cursos, recursos educativos, objetos de aprendizaje, usuarios, y opiniones emitidas por los estudiantes, etc.

SERE fue diseñado para realizar la interacción con estudiantes con el objetivo de permitir que expresen sus opiniones y comentarios libremente acerca de los recursos educativos u objetos de aprendizaje consultados sobre los temas de un curso.

En la Fig. 1 se muestra como el estudiante puede interactuar con el sistema usando diferentes dispositivos con acceso a Internet como son: Tablet, Smartphone, Laptop, o PC.

El usuario ingresa a SERE, usando su cuenta y clave de usuario, a través de la URL (<http://posgradoitc.ddns.net:8000/>) donde se alberga el sistema.

El usuario selecciona el curso y el sistema despliega la lista de temas y los recursos educativos disponibles, permitiendo que el usuario acceda al objeto de aprendizaje que le interesa. Posteriormente, el usuario debe escribir una opinión sobre el OA, la cual se etiqueta con la emoción que el estudiante expresa en ese momento. Cada una de las opiniones de los estudiantes son registradas en una base de datos que contiene diferentes campos como: datos del estudiante, el tema seleccionado del curso, la opinión y evaluación ingresada por el estudiante y por último la fecha y hora en que realizó la opinión.



**Fig. 2.** Proceso del Módulo de Análisis de Sentimientos.

### 3.2. Módulo de análisis de sentimientos (MAS)

El módulo de análisis de sentimientos se encarga de determinar la polaridad (positiva, neutral o negativa) o la emoción centrada en el aprendizaje (aburrido, frustrado, emocionado y comprometido) de una frase, oración o documento.

Anteriormente el módulo de análisis de sentimiento [14] determinaba solamente la polaridad del texto (positiva, neutral o negativa), este módulo se adaptó para reconocer las emociones centradas en el aprendizaje agregando las etiquetas: aburrido, frustrado, comprometido y emocionado. El módulo de análisis de sentimientos se desarrolló usando el clasificador Bernoulli Naive Bayes porque es de los clasificadores más simples y comúnmente utilizados sin embargo utiliza otros tipos de clasificadores como son: Multinomial Naive Bayes, Support Vector Machine, Linear Support Vector Machine, Stochastic Gradient Descent Classifier, and K-Nearest Neighbors.

Para probar el funcionamiento del módulo de análisis de sentimientos, se utilizó un corpus de frases etiquetadas con su respectiva emoción centrada en el aprendizaje las cuales fueron recolectadas de diversas fuentes como: SERE [17], Twitter, plataformas educativas y del corpus TASS [15].

La Fig. 2 muestra el proceso que utiliza el módulo de análisis de sentimiento donde se aprecia que el Corpus de frases es la entrada al proceso que consta de 4 pasos.

Pasos del proceso para análisis de sentimientos:

1. Pre-procesamiento: normaliza las frases contenidas en el corpus. Los pasos de pre-procesamiento son:
  - a) Slang terms: traduce la jerga y emoticones a su equivalente en texto.
  - b) Tokenizador: separa las sentencias en palabras removiendo puntos y signos.
  - c) Stop-words: remueve las palabras innecesarias.
  - d) Stemming: reduce las palabras a su palabra raíz.
2. Transformación: se genera una matriz TF-IDF que calcula la frecuencia de un término (número de veces que aparece un término dado en un documento o conjunto de datos) y la frecuencia inversa de documento (número de documentos en los que aparece un término dado) para cada palabra en el corpus. Esto se conoce como extracción de características de ponderación.
3. Clasificación: define una función para predecir la etiqueta ingresada como entrada.

**Tabla 1.** Opiniones en el Corpus de frases etiquetadas con emoción centrada en el aprendizaje.

Clave	Opinión en Español	Formato	Evaluación
E-15	Me gusto bastante el vídeo	Video	Comprometido
E-64	No me gusto que las voces fueran de España, además los gráficos de la animación están algo feos.	Video	Frustrado
E-70	El video es bueno aunque creo que le falto profundizar mas	Video	Neutral
E-29	Vaya, es algo complejo	Video	Frustrado
E-67	Quizás con un ejemplo quedaría más claro.	Imagen- Texto	Aburrido
E-71	Hubiese sido mejor poner una tabla con sus diferencias y así compararlas y sea más diverso.	Imagen- Texto	Emocionado

4. Evaluación: evalúa el modelo de aprendizaje máquina para predecir la polaridad (positivo, negativo) de un texto de entrada; esto nos ayuda a encontrar un modelo confiable.

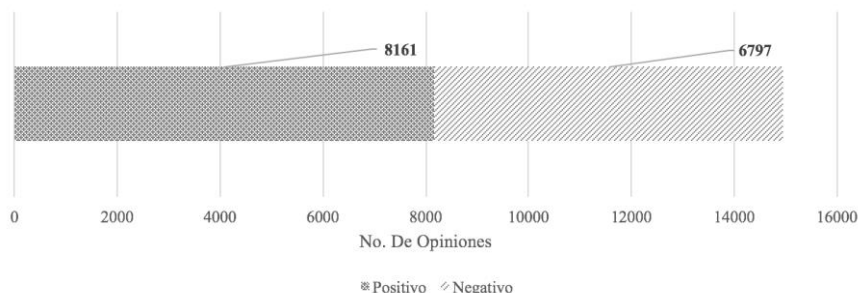
## 4. Resultados

En esta sección, se presentan los resultados obtenidos en los experimentos realizados con el clasificador de frases.

### 4.1. Evaluación del sistema de evaluación de recursos educativos

El primer experimento fue realizado en Agosto 2017, y participaron 53 estudiantes (45 hombres y 8 mujeres) de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Culiacán. Los estudiantes usaron SERE, con el curso de Fundamentos de Programación y los recursos educativos de la unidad 1. Después de estudiar cada subtema en formato de texto e imágenes o multimedia (video), el estudiante registró su opinión con un texto de longitud de 15 a 255 caracteres respecto al contenido del tema estudiado, además, seleccionó una etiqueta para su opinión usando un emoticón que representaba su estado emocional: aburrido, frustrado, neutral, emocionado o comprometido. En la Tabla 1 se muestran algunos ejemplos de las opiniones registradas por los estudiantes que participaron en el estudio.

Con la información obtenida en esta evaluación de SERE, primero se validó con un grupo de profesores que las opiniones de los estudiantes tuvieran relación con la emoción capturada con el emoticón que tenía asociado. Se encontró un 5% de incongruencias y se realizaron las correcciones correspondientes. En este proceso también se detectó que el 20% opiniones escritas eran definiciones de términos básicos tal como algoritmo, computadora, entre otras, por lo que esas opiniones se consideraron con la emoción neutral. Posteriormente se utilizó el Módulo Analizador de Sentimiento para validar las coincidencias con el contenido emocional de los textos, considerando frustrado y aburrido como una emoción negativa y las etiquetas: neutral, emocionado y comprometido como una emoción positiva dado que el Módulo Analizador solo reconocía textos etiquetados como positivos o negativo.



**Fig. 3.** Distribución del corpus por Polaridad.

#### 4.2. Evaluación del modelo de aprendizaje máquina

Una serie de métricas se utilizan para estimar la calidad del algoritmo de clasificación que es parte del modelo de aprendizaje propuesto. En este caso, se usó el método más simple para calcular la efectividad de un clasificador, que es la medida de precisión. Esta calcula el porcentaje de documentos de texto correctamente clasificados sobre el total de documentos a clasificar. Para obtener este valor, se aplicó una técnica de validación cruzada con un conjunto de datos en español con un 90% para los datos de entrenamiento y un 10% para los datos de prueba.

#### 4.3. Corpus de opiniones etiquetados con polaridad

El corpus de frases generado con SERE, contiene tanto etiquetas de polaridad (positivo y negativo) como etiquetas de emociones centradas en el aprendizaje (frustrado, aburrido, comprometido y emocionado).

Actualmente el corpus contiene 14,958 frases, la Fig. 3 muestra la distribución del corpus por polaridad, el cual contiene 8,161 opiniones positivas y 6,797 negativas.

Para el análisis de sentimientos usando el corpus generado con SERE, se utilizaron los siguientes algoritmos de clasificación: Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Support Vector Machine, Linear Support Vector Machine, Stochastic Gradient Descent, and K-Nearest Neighbors (KNN). El clasificador con el puntaje más alto fue Bernoulli Naïve Bayes con una exactitud (*accuracy*) del 83.56%. En la tabla 2 se muestra los valores obtenidos utilizando el clasificador Bernoulli NB.

##### 4.3.1. Corpus de opiniones etiquetadas con emociones centradas en el aprendizaje

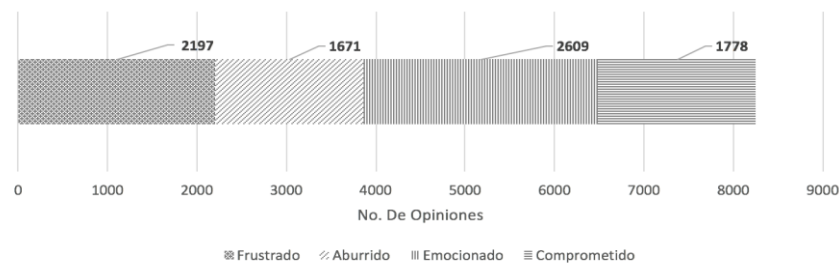
Uno de los principales objetivos del SERE fue recopilar opiniones de los estudiantes para generar un nuevo corpus de opiniones en español para reconocer emociones centradas en el aprendizaje (frustrado, aburrido, emocionado y comprometido) con la finalidad de mejorar el Módulo Analizador de Sentimiento que solo reconoce polaridad, esto es, dos estados emocionales: positivo y negativo. En la Fig. 4 se muestra la distribución de las opiniones recolectadas de septiembre 2017 a marzo 2018.

**Tabla 2.** Valores obtenidos usando el clasificador Bernoulli Naïve Bayes.

	Precisión	Recall	F1-score	Accuracy
Avg/ total	0.84	0.84	0.84	0.835

**Tabla 3.** Valores obtenidos usando el clasificador Bernoulli Naïve Bayes.

	Precisión	Recall	F1-score	Accuracy
Avg/ total	0.58	0.57	0.54	0.574



**Fig. 4.** Distribución del corpus por Emociones Centradas en el Aprendizaje.

El corpus de opiniones etiquetadas con emociones centradas en el aprendizaje se utilizó para entrenar el algoritmo de clasificación Bernoulli Naïve Bayes obteniendo una *accuracy* de 57.40%. En la Tabla 3 se muestran los resultados obtenidos.

## 5. Conclusiones

En este trabajo se generó un corpus de emociones basadas en el aprendizaje utilizando el sistema SERE el cual contiene 9,000 opiniones textuales y cada una de ellas esta etiquetada con una emoción relacionada con el aprendizaje. Además, se trabajó con un módulo de análisis de sentimiento en el cual actualmente trabaja con dos diferentes formas de clasificar una frase de texto.

La primera es utilizando el corpus de emociones centradas en el aprendizaje la cual tiene una precisión de 57.40% utilizando el algoritmo de clasificación Bernoulli Naïve Bayes. La segunda es utilizando el corpus etiquetado con polaridad en el cual se utilizó diferentes algoritmos de clasificación sin embargo la que obtiene mejor precisión es Bernoulli Naïve Bayes con 76.77%.

Por otra parte, el corpus será utilizado posteriormente para que el administrador de un Ambiente Inteligente de Aprendizaje tome decisiones acerca de la pertinencia de los recursos educativos que contiene el sistema y proponga cambios o mejoras a los mismos. Esto ayudará a que los Sistemas Tutores Inteligentes detecten emociones a través de texto y realicen de manera más eficiente el proceso de enseñanza con los estudiantes, ajustando el contenido a las necesidades particulares de cada uno de ellos.

Para trabajos futuros, se está considerando la inclusión de ejercicios para desarrollo de programas Java en el sistema SERE, esto permitirá además crear un banco de

problemas para el aprendizaje de la programación de computadoras, así como ampliar el corpus de opiniones en esta área del conocimiento.

## Referencias

1. D'Mello, S., Jackson, T., Craig, S., Morgan, B., Chipman, P., White, H., Graesser, A.: AutoTutor detects and responds to learners affective and cognitive states. In: Workshop on emotional and cognitive issues at the international conference on intelligent tutoring systems, pp. 306–308 (2008)
2. Ekman, P.: An argument for basic emotions. *Cognition and Emotion*, 6, pp. 169–200 (1992)
3. Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., Graesser, A.: Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive affective states during interactions with three different computer-based learning environments. *Int. J. Hum.-Comput. Stud.*, 68, pp. 223–241 (2010)
4. D'Mello, S.K., Graesser, A.: Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2), pp.145–157 (2012)
5. Pekrun, R.: The impact of emotions on learning and achievement: Towards a theory of cognitive/motivational mediators. *Applied Psychology*, 41(4), pp. 359–376 (1992)
6. Pekrun, R., Goetz, T., Titz, W., Perry, R.P.: Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research. *Educational psychologist*, 37(2), pp. 91–105 (2002)
7. Feldman, R.: Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), pp. 82–89 (2013)
8. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp. 1093–1113 (2014)
9. Altrabsheh, N., Gaber, M., Cocea, M.: SA-E: Sentiment Analysis for Education. *Frontiers in Artificial Intelligence and Applications*, 255 (2013)
10. Ortigosa, A., Martín, J.M., Carro, R.M.: Sentiment analysis in Facebook and its application to e-learning. *Comput. Hum. Behav.*, 31, pp. 527–541 (2014)
11. Rowe, A.D.: Feelings About Feedback: The Role of Emotions in Assessment for Learning. In: *Scaling up Assessment for Learning in Higher Education*, Carless, D., Bridges, S.M., Chan, C.K.Y., Glofcheski, R., Eds. Singapore: Springer Singapore, pp. 159–172 (2017)
12. Altrabsheh, N., Cocea, M., Fallahkhair, S.: Learning Sentiment from Students' Feedback for Real-Time Interventions in Classrooms. In: *Adaptive and Intelligent Systems: Third International Conference, (ICAIS'14), Proceedings*, A. Bouchachia, Cham: Springer International Publishing, pp. 40–49 (2014)
13. Munezero, M., Montero, C.S., Mozgovoy, M., Sutinen, E.: Exploiting sentiment analysis to track emotions in students' learning diaries. In: *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, pp. 145–152 (2013)
14. Barrón-Estrada, M.L., Zatarain-Cabada, R., Oramas-Bustillos, R., González-Hernández, F.: Sentiment Analysis. In: *An Affective Intelligent Tutoring System, IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, Timisoara, pp. 394–397 (2017)
15. Villena-Román, J., Martínez-Cámara, E., Lana-Serrano, S., González-Cristóbal, J.C.: TASS-Workshop on Sentiment Analysis at SEPLN TASS. *Taller de Análisis de Sentimientos en la SELPLN*, pp. 37–44 (2013)
16. Tellez, E.S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Siordia, O.S., Villaseñor, E. A.: A case study of Spanish text transformations for twitter sentiment analysis. *Expert Systems with Applications*, pp. 457–471 (2017)
17. Barrón-Estrada, M.L., Zatarain-Cabada, R., Oramas-Bustillos, R., Ramírez-Ávila, S.L.: Building a Corpus of Phrases Related to Learning for Sentiment Analysis. *Research in Computing Science*, 146, pp. 17–26 (2017)



## Interfaz de lenguaje natural para deducción de información almacenada en ontologías

Alejandro Solís-Sánchez, Rogelio Florencia-Juárez,  
Juan Carlos Acosta Guadarrama, Francisco López-Orozco

Universidad Autónoma de Ciudad Juárez, Chihuahua,  
México

al160509@alumnos.uacj.mx,  
{rogelio.florencia, juan.acosta, francisco.orozco}@uacj.mx

**Resumen.** Las interfaces de lenguaje natural a ontologías permiten consultar datos de un dominio específico, almacenados en ontologías diseñadas mediante el uso de lenguajes de ontología, como el lenguaje OWL. Para acceder a estos datos, se requieren lenguajes de consulta complejos como SPARQL, utilizados solo por usuarios especializados. Las interfaces de lenguaje natural a ontologías traducen consultas formuladas en lenguaje natural a consultas SPARQL. En este trabajo, se describe la arquitectura de una interfaz de lenguaje natural a ontologías. Recibe una consulta formulada por los usuarios en un dispositivo Android utilizando su voz. La consulta se transforma en texto y se envía a un servidor Java Server Pages (JSP). El núcleo de nuestra interfaz, que se ejecuta en un servidor JSP, recibe la consulta del usuario convertida en texto y, mediante técnicas de procesamiento de lenguaje natural, la transforma en una consulta SPARQL, que se utiliza para extraer de la ontología la información solicitada por los usuarios. Para traducir consultas en lenguaje natural a consultas SPARQL, la interfaz utiliza técnicas de procesamiento del lenguaje natural como Tokenización, Lematización y etiquetado gramatical, así como la extracción del conocimiento almacenado en la ontología a consultar y diálogos de aclaración para responder las preguntas del usuario.

**Palabras clave:** interfaces de lenguaje natural a ontologías, SPARQL, OWL, Web semántica.

## Natural Language Interface for Information Deduction Stored in Ontologies

**Abstract.** Natural language interfaces to ontologies allow querying data of a specific domain, stored in ontologies designed by using ontology languages such as the Ontology Web Language. To access this data, complex query languages such as SPARQL, used by only specialized users, are required. Natural language interfaces to ontologies translate queries formulated in natural language to SPARQL queries. In this paper, the architecture of a natural language interface to ontologies is described. It receives a query formulated by users on an Android device using their voice. The query is transformed into text and it is sent to a Java

Server Page (JSP) server. The core of our interface, running on the JSP server, receives the user's query converted to text and, through natural language processing techniques, it transforms it into a SPARQL query, which is used to extract from the ontology to query, the information requested by users. To translate natural language queries to SPARQL queries, the interface uses natural language processing techniques such as Tokenization, Lemmatization, and Part of Speech tagging, as well as, extraction of knowledge stored in the ontology to be queried, and clarification dialogues to answer user questions.

**Keywords:** natural language interfaces to ontologies, SPARQL, OWL, semantic web.

## **1. Introducción**

La Web Semántica es una extensión de la web actual [1], donde la información recibe un significado bien definido, que puede ser entendido por computadoras y humanos. La arquitectura de la Web Semántica que fue establecida por Tim Berners-Lee (considerado como el creador de la Web), hace uso de ontologías como uno de sus principales componentes. Una ontología es una especificación explícita de una conceptualización. Las ontologías permiten comprender mejor la estructura del conocimiento, ya que muestran los conceptos y las relaciones que existen entre ellos.

En la Web Semántica la información es almacenada haciendo uso de las ontologías. Dichas ontologías son representadas a través del lenguaje OWL (considerado una extensión del lenguaje Resource Description Framework RDF) [2]. Consultar el conocimiento almacenado en una ontología requiere de conocimientos avanzados diseñados para tal propósito, un ejemplo es SPARQL (Protocol and RDF Query Language) [3], conocimientos que la mayoría de las personas no posee [4]. Por tal motivo se han implementado interfaces que reciben una consulta en lenguaje natural y la transforman a una consulta SPARQL con la cual extraen la información solicitada por los usuarios.

La Web Semántica necesita de interfaces para consultar ontologías en un lenguaje más cercano al que los humanos conocemos y entendemos. Esto nos facilita la interacción con sitios o aplicaciones basados en la arquitectura de la misma.

En este trabajo se propone la arquitectura para una interfaz que, mediante consultas en lenguaje natural, permita buscar información almacenada en una ontología, utilizando un dispositivo móvil para su funcionamiento. En la Sección 2 se agrega una descripción de trabajos relacionados existentes, en la Sección 3 se agrega la descripción de la propuesta de este trabajo, la Sección 4 describe la implementación y en la Sección 5 las conclusiones.

## **2. Trabajos relacionados**

Existen diferentes arquitecturas propuestas para la implementación de interfaces de lenguaje natural a ontologías, utilizan diferentes técnicas de procesamiento de lenguaje natural y lenguajes formales de consulta. A continuación, se describe la arquitectura de

algunas interfaces de lenguaje natural a ontologías como son FREYA [5], Gingseng [6], QuestIO [7], ORAKEL [8] y AquaLog [9].

## **2.1. FREyA**

Es una interfaz de lenguaje natural (ILN) que puede usarse para consultar ontologías en formato RDF/OWL. FREyA (Feedback Refinement and Extended Vocabulary Aggregation) mapea a SPARQL una consulta en lenguaje natural en forma de pregunta, luego se ejecuta en un repositorio RDF/OWL y devuelve una respuesta [5]. FREyA usa el conocimiento disponible en la ontología para identificar los términos que se encuentren en la consulta. A dichos términos se les llama (OC). Si existe ambigüedad en los términos, FREyA genera un diálogo de clarificación en el que el usuario elige una de varias opciones. En el diálogo de clarificación el usuario elige una opción que es almacenada y usada para entrenar al sistema.

A partir de los OC FREyA genera un conjunto de tripletas que convierte a consultas SPARQL para buscar la respuesta en la ontología. Para mostrar los resultados al usuario, se identifica primero el tipo de respuesta [5]. Para el tipo de respuesta realiza un análisis sintáctico y una búsqueda en la ontología para identificar el focus de la pregunta. Un focus es una palabra o secuencia de palabras que definen una pregunta y la desambiguan indicando lo que está buscando [10]. La presentación de los resultados de FREyA incluye un grafo para su visualización.

## **2.2. Gingseng**

Ginseng (Guided Input Natural Language Search Engine) es otra ILN que se basa en una gramática de preguntas que se extiende dinámicamente por la estructura de una ontología para guiar a los usuarios en la formulación de consultas en un idioma aparentemente similar al inglés. Basándose en la gramática, Ginseng traduce las consultas al lenguaje SPARQL, que permite su ejecución [6].

Proporciona un acceso de consulta a cualquier base de conocimiento OWL. Guía al usuario a formular consultas por lo que no es necesario interpretarlas (lógica o sintácticamente) y no utiliza ningún vocabulario predefinido. Ginseng sólo conoce el vocabulario definido por las ontologías consideradas actualmente y el usuario tiene que seguirlo [6]. Esto puede limitar las posibilidades del usuario en general, pero asegura que todas las consultas puedan ser respondidas [6].

La arquitectura de Ginseng consta de tres partes: una gramática multinivel, un parser incremental y una capa de acceso a la ontología. La gramática multinivel consta de una parte estática que especifica las estructuras de oraciones de consulta generalmente posibles y una parte dinámica que se genera a partir de las ontologías utilizadas. Las reglas gramaticales estáticas proporcionan las estructuras y frases básicas para las preguntas en inglés. Las reglas gramaticales dinámicas se generan a partir de cada ontología cargada en Ginseng y son utilizadas para extender la parte de las gramáticas estáticas [6].

La gramática completa es utilizada por el parser incremental primero para proporcionar alternativas al usuario durante el ingreso de consultas, y segundo para almacenar información sobre cómo construir consultas SPARQL. Finalmente, el framework Jena es utilizado para la capa de acceso a las ontologías. Cuando se completa

la ejecución de una consulta, Ginseng muestra la consulta SPARQL generada y los resultados para el usuario [6].

Ginseng, a diferencia de FREyA, su principal característica es que las consultas que realiza el usuario son controladas mediante una gramática basada en los conocimientos almacenados en la ontología. Esto da lugar a una de sus ventajas e igualmente a su desventaja, ya que la gramática le da el control sobre lo que escribe el usuario y así evitar consultas no válidas, pero a la vez limita las consultas que el usuario puede realizar.

### **2.3. QuestIO System**

QuestIO (Question-Based Interface to Ontologies) es una ILN para acceder a información estructurada, es independiente del dominio y es fácil de usar sin formación [7]. Es de dominio abierto (o personalizable a nuevos dominios con muy poco costo), con el vocabulario no predefinido, es decir, que se deriva automáticamente de los datos existentes en la base de conocimientos [7]. El sistema trabaja convirtiendo consultas en lenguaje natural a consultas formales de tipo SeRQL [11] (aunque se pueden usar otros lenguajes de consulta).

El proceso de funcionamiento de QuestIO empieza por procesar la ontología del dominio, creando de manera automática un diccionario léxico a partir del conocimiento obtenido. A partir de la ontología el diccionario es capaz de identificar menciones de clases, propiedades, instancias y valores de propiedad asociados con las instancias [11].

Cuando QuestIO recibe una consulta, el sistema realiza los siguientes pasos:

- Realiza un análisis lingüístico que consiste en un análisis morfológico del texto por medio de un tokenizador y herramienta de etiquetado.
- La segunda fase es ejecutar el diccionario ontológico creado al iniciar el sistema sobre el texto de la consulta. Esto crea anotaciones para todas las menciones que el diccionario pudo identificar clases, propiedades, instancias y valores de propiedades de tipos de datos.
- Inicia un proceso de transformación iterativo para convertir el texto de entrada en una consulta formal. Primero separa el texto de entrada en tokens, determinando la parte del discurso de cada uno y agregando anotaciones con su raíz morfológica. Después trata de identificar en el texto de entrada las menciones de los recursos de la ontología para poder generar una consulta formal en SeRQL.
- Finalmente ejecutar la consulta en la base de conocimiento y desplegar los resultados.

QuestIO es una ILN que a diferencia de FREyA y Ginseng, utiliza el lenguaje de consultas SeRQL pudiendo ser una desventaja ya que el estándar oficial para consulta de ontologías en la Web Semántica es el lenguaje SPARQL. Otra desventaja es que depende de una ontología bien diseñada para su funcionamiento, lo cual generalmente requiere de un especialista del dominio para el diseño de la ontología, lo que lo hace menos accesible para usuarios no especializados.

## 2.4. ORAKEL

ORAKEL es una ILN a base de conocimientos que convierte las preguntas a forma lógica [8] ya que para la representación del conocimiento utiliza el lenguaje f-logic [12]. Recibe como entrada una consulta de lenguaje natural que es convertida a una fórmula lógica de primer orden. Después la fórmula lógica es transformada al lenguaje de consulta específico, que puede ser SPARQL o el lenguaje de consultas de f-logic [13]. ORAKEL tiene los siguientes componentes principales:

1. Léxico del dominio y léxico general, ambos creados por el experto del dominio.
2. Base de conocimiento, compuesta por la ontología del dominio.
3. FrammeMapper, es un experto en el dominio que debe conocer la base de conocimientos subyacente.
4. Intérprete de consultas (*Query interpreter*), construye una consulta formulada por el usuario a forma lógica con respecto a los predicados del dominio.
5. Convertidor de la consulta (*Query converter*). Está implementado en lenguaje Prolog. Este componente recibe consultas en forma lógica y las traduce al lenguaje de la base de conocimientos (f-logic).
6. Generación de respuesta (*Answer generation*).

A diferencia de las otras ILN expuestas en este trabajo, ORAKEL es compatible también con el lenguaje f-logic para la representación del conocimiento. Su principal ventaja es que fue diseñado para portar ILN's entre dominios de manera eficiente. La desventaja es que requiere del conocimiento de un experto del dominio para la generación del léxico.

## 2.5. AquaLog

Es un sistema de pregunta-respuesta portable que toma como entrada consultas expresadas en lenguaje natural y una ontología y devuelve respuestas extraídas del marcado semántico compatible con la ontología disponible [9]. La arquitectura de AquaLog se puede caracterizar como un modelo en cascada en el que una consulta en lenguaje natural se traduce a un conjunto de representaciones intermedias basadas en tripletas. Dichas tripletas son traducidas a tripletas compatibles con la ontología. El modelo tripletas que usa AquaLog es de la forma (sujeto, predicado, objeto) [9]. Los módulos principales de AquaLog son el componente lingüístico y el servicio de similitud de relación.

La función del componente lingüístico es convertir la consulta de lenguaje natural a una consulta en tripletas. AquaLog utiliza la infraestructura y los recursos de GATE [14] para analizar la pregunta como parte del componente lingüístico. GATE devuelve un conjunto de anotaciones sintácticas asociadas con la consulta de entrada. Estas anotaciones incluyen información sobre oraciones, tokens, sustantivos y verbos. AquaLog extiende el conjunto de anotaciones devueltas por GATE, identificando términos, relaciones, indicadores de preguntas (qué/quién/cuándo/etc.) y patrones o tipos de preguntas.

AquaLog presenta una solución en la que se combinan diferentes estrategias para dar sentido a una consulta en lenguaje natural con respecto al universo del discurso cubierto

por la ontología. Utilizando el framework GATE le da capacidad para mejorar el procesamiento de las consultas en lenguaje natural.

Las interfaces de lenguaje natural analizadas anteriormente utilizan ontologías para almacenar conocimiento, el cual se almacena en forma de archivos xml y se extrae mediante un lenguaje de consulta. A excepción de QuestIO (que utiliza SeRQL), las demás interfaces utilizan el lenguaje SPARQL que es el estándar oficial para usarse en la Web Semántica.

### **3. Arquitectura propuesta**

La interfaz que actualmente se tiene en desarrollo, recibe una consulta formulada en lenguaje natural por los usuarios en idioma inglés, la traduce al lenguaje de consultas de ontologías SPARQL [3] y mediante ésta, extrae de la ontología del usuario la información solicitada. Su arquitectura está dividida en dos módulos principales: a) el módulo generador de conocimiento (MGC) y b) el módulo de la interfaz (MI). El MGC se encarga de generar el conocimiento que el MI necesita para responder a las consultas formuladas en lenguaje natural por los usuarios. En la Sección 3.1 se describe el MGC y en la Sección 3.2 se describe el MI. En la Fig. 1 se presenta la arquitectura general de la interfaz propuesta.

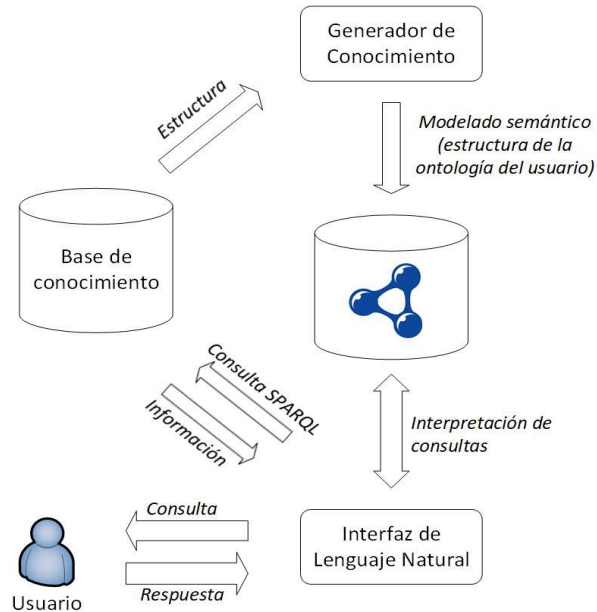
#### **3.1. Módulo generador de conocimiento**

Para que el MI sea capaz de responder consultas formuladas en lenguaje natural, es necesario proveerla de conocimiento acerca de la estructura de la ontología que el usuario desea consultar y acerca del dominio lingüístico conversacional. Con el fin de facilitar su portabilidad a diferentes ontologías del usuario, el MGC es el encargado de generar este conocimiento, buscando minimizar la intervención del usuario al configurar el MI.

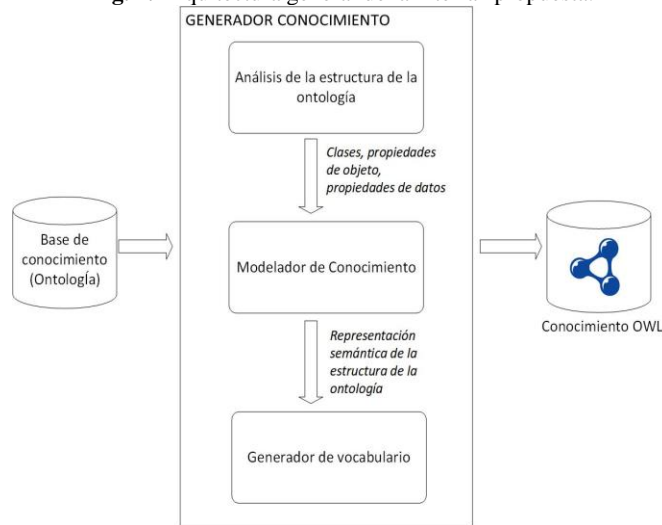
Como primer paso, el usuario indica al MGC la ontología que desea consultar. Posteriormente, el MGC accede a la ontología y analiza su estructura, es decir, identifica las clases, propiedades de objeto, propiedades de datos y la manera en que estos elementos se encuentran relacionados. Para este fin, se utilizó el lenguaje de consulta SPARQL, en el lenguaje de ontologías OWL [2]. A través de consultas formuladas en SPARQL se pueden identificar los elementos que componen la estructura de una ontología y se puede acceder a la información almacenada, generalmente modelada como individuos, es decir, como instancias de las clases en la ontología.

Como siguiente paso, los elementos identificados son modelados semánticamente utilizando una representación semántica definida a priori para este propósito. La representación semántica se diseñó primeramente en el software Protégé [15]. Posteriormente, para integrar la estructura de esta representación semántica en el MGC, se utilizó el framework Apache Jena [16], el cual permite gestionar ontologías desde el lenguaje de programación Java, en el cual está diseñada la interfaz propuesta.

A continuación, cada uno de los nombres de los elementos modelados son adicionados con sinónimos, así como con palabras que compartan el mismo lema. Esto se realiza con el fin de proveer al MI de conocimiento lingüístico acerca del dominio



**Fig. 1.** Arquitectura general de la interfaz propuesta.

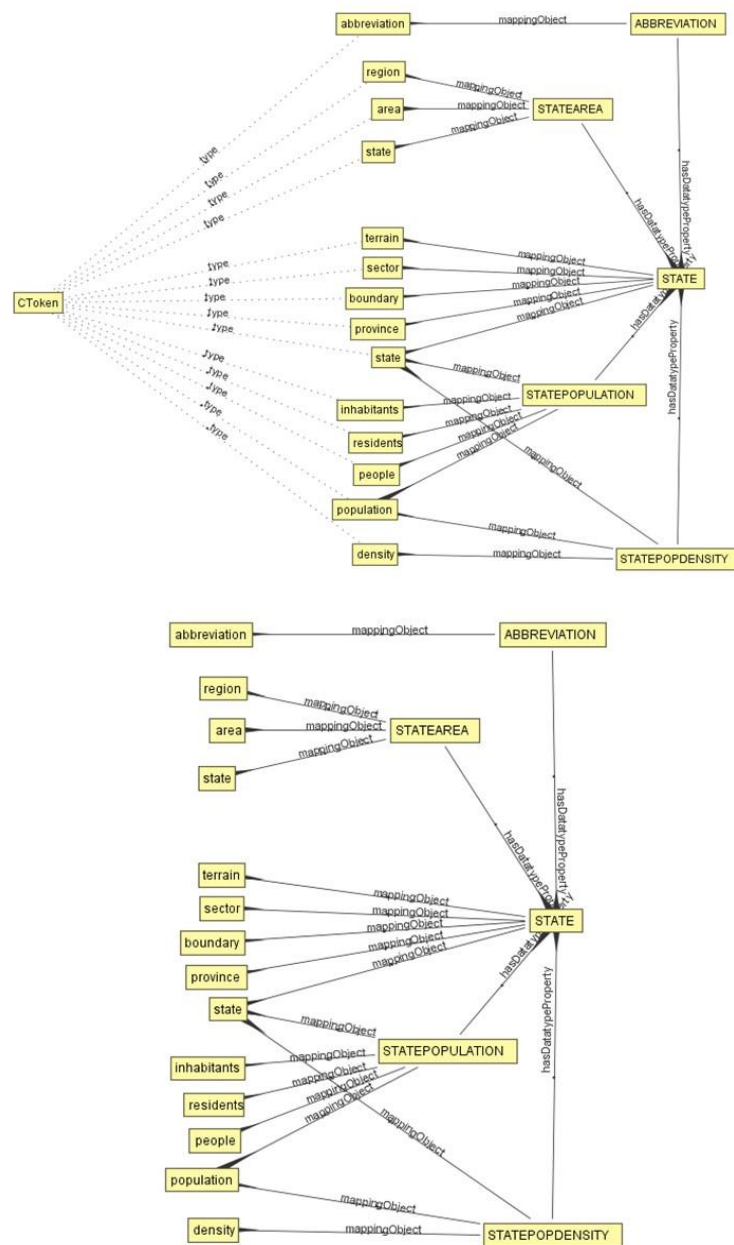


**Fig. 2.** Arquitectura del módulo generador de conocimiento.

de la ontología del usuario. El vocabulario generado también es agregado al modelado semántico.

Por último, después de generar el modelado semántico, el MGC lo almacena en una ontología, la cual es creada dinámicamente. Esta ontología generada por el MGC, contiene el conocimiento que el MI requiere para responder a consultas formuladas en lenguaje natural por los usuarios. En la Fig. 2 se presenta la arquitectura

correspondiente al MGC y en la Fig. 3 se muestra un fragmento del modelado semántico generado.



**Fig. 3.** Fragmento del modelado semántico generado.



El fragmento presentado fue generado a partir de la ontología Geography.owl. Utilizada en la evaluación de la interfaz FREyA por Damjanovic [5]. A su vez, esta ontología fue generada a partir de la base de datos deductiva de Raymond J. Mooney [17], la cual contiene información acerca de la geografía de Estados Unidos y fue distribuida como datos de ejemplo en Turbo Prolog 2.0. En la Fig. 4 se muestra la estructura de la ontología Geography.owl.

La representación semántica de la Fig. 3 está conformado por las clases CClass, CObjectProperty, CDatatypeProperty y CToken. Además de las propiedades de objeto hasDatatypeProperty, mappingObject, entre otras.

Las clases State, Lake, City, Capital, Lake, LoPoint, Mountain, HiPoint, River y Road, mostradas en la Fig. 4, son modeladas en la Fig. 3 como individuos de la clase CClass.

Las propiedades de objeto Borders, isCityOf, hasCity, isCapitalOf, hasCapital, isLakeOf, hasLake, isLowestPointOf, hasLowestPoint, isMountainOf, hasMountain, isHighestPointOf, hasHighestPoint, runsThrough, hasRiver, passesThrough y hasRoad, mostradas en la Fig. 4, son modeladas en la Fig. 3 como individuos de la clase CObjectProperty.

Las propiedades de datos statePopulation, stateArea, abbreviation, statePopDensity, cityPopulation, lakeArea, loElevation, height, hiElevation, length y number, mostradas en la Fig. 4, son modeladas en la Fig. 3 como individuos de la clase CDatatypeProperty.

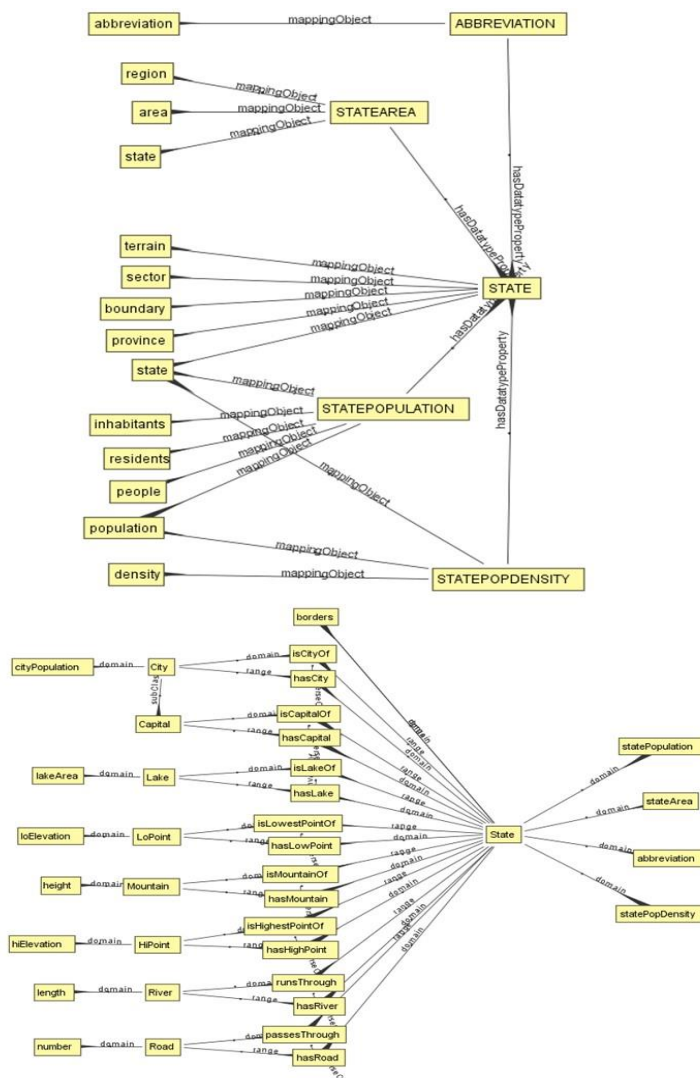
La clase CToken es utilizada para modelar el vocabulario de la interfaz. Considerando la Fig. 3, si el usuario introduce la palabra “population”, el MI sabrá que posiblemente se está refiriendo a los individuos STATE (CClass), STATEPOPULATION (CObjectProperty) o STATEPOPDENSITY (CObjectProperty). De esta manera el MI utiliza el conocimiento generado para interpretar las consultas de los usuarios y generar la consulta SPARQL.

### **3.2. Módulo de la interfaz**

El MI tiene como objetivo traducir la consulta formulada en lenguaje natural por el usuario a una consulta SPARQL, con la cual extraerá de la ontología del usuario la información solicitada. Su funcionamiento es mostrado en la Fig. 5.

El primero paso efectuado por el MI para generar la consulta SPARQL, es realizar a la consulta introducida por el usuario, una fase de procesamiento de lenguaje natural. Esta fase consiste en separar la consulta del usuario en un conjunto de palabras o tokens aislados. Posteriormente, estos tokens son etiquetados de acuerdo a su función gramatical dentro de la consulta del usuario. Por último, se obtiene el lema de cada uno de los tokens. Esta fase es realizada utilizando Freeling [18], la cual es una suite de herramientas para el procesamiento de lenguaje natural que ofrece soporte para diversos idiomas, entre ellos, inglés y español. Suponiendo que el usuario introduce la consulta “What is the population density of Wyoming?”, el resultado de esta fase es como se muestra en la Tabla 1.

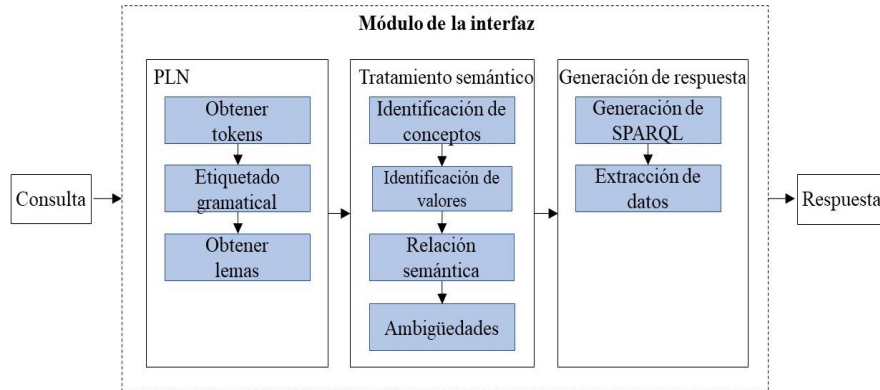
El segundo paso que realiza el MI es un tratamiento semántico. En esta fase, se analiza cada token que haya sido etiquetado gramaticalmente como un sustantivo (noun) o un adjetivo (adjective). El análisis consiste en identificar si algún token mapea a algún concepto, es decir, a algún individuo de la clase CToken de la ontología



**Fig. 4.** Estructura de la ontología Geography.owl.

generada por el MGC (tal como se mostró en el ejemplo de la palabra population al final de la Sección 3.1), lo cual se realiza a través de una coincidencia exacta de caracteres.

Mediante el modelado presentado en la Fig. 3, el MI identifica si el mapeo hace referencia a una clase, propiedad de objeto o propiedad de datos. Si alguno de estos tokens no mapea a algún elemento en la ontología, se realiza una búsqueda en la ontología del usuario para identificar si se trata de algún valor. En la consulta del ejemplo anterior, Wyoming se considera como un valor, ya que el MI identifica que es



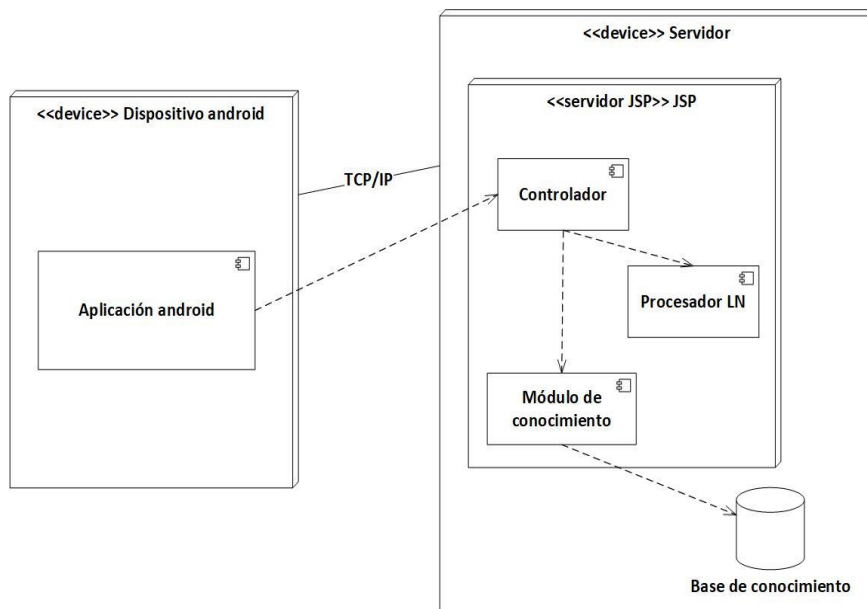
**Fig. 5.** Funcionamiento del módulo de la interfaz.

**Tabla 1.** Ejemplo del resultado del procesamiento de lenguaje natural.

Token	Lema	PoS	Significado PoS
What	What	WP	pos= pronoun type= interrogative
is	Be	VBZ	pos= verb vform =personal person=3
the	the	DT	pos= determiner
population	population	NN	pos=noun type=common num=singular
density	density	NN	pos=noun type=common num=singular
of	of	IN	pos=preposition
Wyoming	wyoming	NP	pos=noun type=proper
?	?	Fit	pos=punctuation type=questionmark punctenclose=close

un individuo instanciado a partir de la clase State. Es importante mencionar que identificar correctamente los mapeos y los valores, es esencial para generar la consulta SPARQL. En la Tabla 2 se presenta el resultado de este paso.

Posteriormente, se intenta identificar si existe alguna relación semántica entre algunos de los tokens. Esto se determina si entre sus mapeos existe alguno en común, además de seguir algunas reglas establecidas a priori. En caso de ser afirmativo, ambos tokens son combinados en un token compuesto, conservando solamente los mapeos coincidentes o relacionados y desechando el resto de sus mapeos. Esto se realiza con el fin de simplificar el proceso de generación de la consulta SPARQL. Como se puede ver



**Fig. 5.** Funcionamiento del módulo de la interfaz.

en la Tabla 2, los tokens *population* y *density* tienen en común el mapeo a *STATEPOPDENSITY*, por lo cual, el MI los combina en un solo token.

Posteriormente, el MI procesa los valores identificados. Si un valor pertenece a más de una clase podría causar un problema de ambigüedad. En este caso, Wyoming pertenece a dos clases, *State* (*wyoming*) y *City* (*wyomingMi*), por lo que se intenta resolver la ambigüedad identificando la(s) clase(s) que concuerde(n) con las clases de los demás elementos de la consulta. Como el MI identificó en base a la Fig. 3 que *STATEPOPDENSITY* es una propiedad de datos que pertenece a *STATE*, se descarta la clase *City*. En caso de que el MI no pueda resolver la ambigüedad, muestra un diálogo de clarificación al usuario para que éste la resuelva. En la Tabla 3 se presenta el resultado de esta fase de tratamiento semántico.

El tercer paso es generar la consulta SPARQL con los elementos identificados después de la fase de tratamiento semántico, mostrados en la Tabla 3 y utilizar esta consulta para extraer de la ontología del usuario la información solicitada. Para generar la consulta SPARQL se sigue una serie de reglas definidas a priori.

Por ejemplo, en base a la Fig. 3 y a la Tabla 3, el MI identificó que *STATEPOPDENSITY* es una propiedad de datos relacionada con la clase *STATE*.

Estos elementos son identificados en la ontología del usuario por el MI como `<http://www.mooney.net/geo#statePopDensity>` y como `<http://www.mooney.net/geo#State>`. Además, Wyoming es un individuo de la clase *State*. Por tal motivo, el MI genera la siguiente consulta SPARQL, donde la variable que forma parte de la cláusula *Select* se forma a partir del token combinado *population density*, el cual es transformado `?population_density`. El resultado mostrado por el MI utilizando la consulta SPARQL generada se muestra en la Fig. 6.

**Tabla 2.** Ejemplo de identificación de mapeos y valores.

Token	Lema	PoS	Significado PoS
What	what	WP	
Is	Be	VBZ	
The	the	DT	
population	population	NN	Token= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#population">http://www.uacj_nlp.com/schema/nlp.OWL#population</a> Class= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty">http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty</a> Object= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPULATION">http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPULATION</a> Token= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#population">http://www.uacj_nlp.com/schema/nlp.OWL#population</a> Class= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty">http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty</a> Object= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPULATION">http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPULATION</a> Token= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#citypopulation">http://www.uacj_nlp.com/schema/nlp.OWL#citypopulation</a> Class= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty">http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty</a> Object= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CITYPOPULATION">http://www.uacj_nlp.com/schema/nlp.OWL#CITYPOPULATION</a>
density	density	NN	Token= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#density">http://www.uacj_nlp.com/schema/nlp.OWL#density</a> Class= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty">http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty</a> Object= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPDENSITY">http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPDENSITY</a> Token= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#population">http://www.uacj_nlp.com/schema/nlp.OWL#population</a> Class= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty">http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProperty</a> Object= <a href="http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPDENSITY">http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOPDENSITY</a>
of	of	IN	
Wyoming	wyoming	NP	Individual= <a href="http://www.mooney.net/geo#wyoming">http://www.mooney.net/geo#wyoming</a> Class= <a href="http://www.mooney.net/geo#State">http://www.mooney.net/geo#State</a> Label= <a href="http://www.mooney.net/geo#wyoming">http://www.mooney.net/geo#wyoming</a> @en Individual= <a href="http://www.mooney.net/geo#wyomingMi">http://www.mooney.net/geo#wyomingMi</a> Class= <a href="http://www.mooney.net/geo#City">http://www.mooney.net/geo#City</a> Label= <a href="http://www.mooney.net/geo#wyoming">http://www.mooney.net/geo#wyoming</a> @en
?	?	Fit	

#### 4. Implementación

En esta sección se describe la implementación de la ILN propuesta en este trabajo. Se describen los componentes de software y los dispositivos de hardware donde son alojados.

La ILN se basa en una arquitectura de tipo cliente-servidor, utiliza una aplicación Android como interfaz gráfica y un servidor de aplicaciones donde se realiza el procesamiento de las consultas. El servidor funciona mediante Java Server Pages (JSP) [19] como tecnología para su implementación por lo que utiliza Apache Tomcat [20] que es un contenedor de aplicaciones web basado en Java.

El diagrama de despliegue de la Fig. 7 muestra los componentes de software y los dispositivos de hardware utilizados. El sistema implementa los siguientes componentes de software: una aplicación Android, un controlador de eventos de interacción entre los

**Tabla 3.** Ejemplo del resultado de la fase de tratamiento semántico.

Token	Lema	PoS	Significado PoS
What	What	WP	
is	Be	VB Z	
the	The	DT	
populati on density	populati on density	NN	Token=http://www.uacj_nlp.com/schema/nlp.OWL#population Class=http://www.uacj_nlp.com/schema/nlp.OWL#CDatatypeProp erty Object=http://www.uacj_nlp.com/schema/nlp.OWL#STATEPOP <b>DENSITY</b>
of	Of	IN	
Wyomin g	Wyomin g	NP	Individual=http://www.mooney.net/geo#wyoming Class=http://www.mooney.net/geo#State Label=wyoming@en
?	?	Fit	

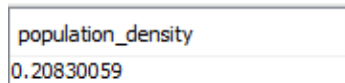
```

Select ?population_density
Where {
    <http://www.mooney.net/geo#statePopDensity>
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://www.w3.org/2002/07/owl#DatatypeProperty> .

    <http://www.mooney.net/geo#wyoming>
    <http://www.mooney.net/geo#statePopDensity>
    ?population_density .

    <http://www.mooney.net/geo#wyoming>
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://www.mooney.net/geo#State>
}

```



population\_density  
0.20830059

**Fig. 6.** Resultado de la consulta “What is the population density of Wyoming” utilizando la ontología Geography.owl.

componentes, una base de conocimientos, un componente para el procesamiento de lenguaje natural y otro para generar y ejecutar consultas SPARQL en la base de conocimiento.

El proceso general para cada consulta de entrada es de acuerdo con los siguientes pasos:

1. La aplicación Android recibe como entrada la consulta en forma de voz, la convierte a texto y la envía al controlador para su procesamiento.
2. El controlador recibe la consulta de texto y la envía al procesador LN.
3. El procesador LN recibe la consulta de texto y a partir de esta obtiene los tokens, etiquetas gramaticales y lemas. Finalmente envía al controlador un conjunto de tokens, etiquetas gramaticales y lemas.

4. Una vez que el controlador recibe el conjunto de tokens etiquetas y lemas, lo envía al módulo de conocimiento.
5. El módulo de conocimiento se encarga de generar consultas SPARQL y ejecutarlas en la ontología para obtener la respuesta. Por último, envía la respuesta al controlador.
6. El controlador envía la respuesta a la aplicación Android para mostrarla al usuario.

Para la implementación de cada componente se utilizaron varias herramientas de desarrollo.

La aplicación Android fue implementada mediante el entorno de desarrollo integrado Android Studio [21], utilizando librerías nativas para convertir consultas de voz a texto. Se puede utilizar en dispositivos que utilicen Android como sistema operativo.

El controlador está implementado mediante la tecnología JSP [22]. Recibe y realiza peticiones a los otros componentes para gestionar todo el proceso del sistema, desde recibir la consulta hasta enviar la respuesta a la aplicación Android.

El procesador LN está implementado en lenguaje Java [23]. Después de revisar varias herramientas de procesamiento de lenguaje natural como la librería de código abierto Freeling [18], la librería Apache OpenNLP [24] y la herramienta Stanford CoreNLP [25], se eligió Freeling para implementar este componente ya que, a diferencia de las otras tiene mejor soporte para los idiomas inglés y español.

La base de conocimiento es almacenada en forma de ontología y en archivos de tipo OWL [2] y RDF [26], para acceder a ella se utiliza el lenguaje de consulta SPARQL. El módulo de conocimiento fue implementado utilizando el lenguaje de programación Java y Apache Jena [16] que es un framework Java para la consulta de bases de conocimientos almacenadas en ontologías RDF y OWL utilizando SPARQL, y que permite gestionar la ontología en tiempo de ejecución.

La ILN permite seleccionar la ontología que se desea consultar previamente creada. Utilizando el framework Jena la ILN accede a la ontología y analiza su estructura para posteriormente responder las consultas en lenguaje natural de los usuarios.

## **5. Conclusiones**

Con la aparición de la Web Semántica el uso de ontologías ha tomado demasiada importancia, ya que se proponen utilizar como medio de representación de conocimiento con el objetivo de facilitar la interoperabilidad de la información en la web. Las interfaces de lenguaje natural a ontologías son excelentes herramientas para acceder al conocimiento almacenado en ontologías. Su desarrollo podría facilitar a los usuarios la localización de recursos, la comunicación entre aplicaciones informáticas, la búsqueda de información, entre otras actividades realizadas en la Web.

En este trabajo se propone la arquitectura de una ILN que utiliza bases de conocimiento basadas en ontologías, la cual recibe consultas de voz desde un dispositivo Android, las convierte a texto, aplica técnicas de procesamiento de lenguaje natural y de representación de conocimiento para generar las consultas SPARQL correspondientes y extrae información de la ontología.

A pesar de que el modelado y el procesamiento semántico que realiza la ILN le han permitido responder a consultas formuladas en lenguaje natural por los usuarios, se han detectado áreas de oportunidad.

Como trabajos futuros a corto plazo, se propone mejorar el esquema de representación de conocimiento, así como el procesamiento semántico de la ILN. También se propone fortalecer el procesamiento de lenguaje natural para que la ILN sea capaz de responder consultas que integren comparaciones, negaciones, fechas y números. Posteriormente, se propone evaluar el desempeño de la ILN contra FREyA utilizando la ontología *Geography.owl* y sus respectivos corpus de consultas llamados *Geoquery 250* y *Geoquery 880*.

Como trabajos futuros a largo plazo, se propone abordar problemas complejos relacionados con el lenguaje natural, tales como anáforas y posteriormente, elipsis intersentencial. La anáfora es cuando se hace referencia a una entidad mencionada anteriormente y su resolución es importante para responder a preguntas que se refieren a la misma entidad en diferentes formas. La elipsis ocurre cuando se omiten una o más palabras en una oración y que son sobrentendidas debido a que ya se han mencionado antes.

**Agradecimientos.** Agradecemos a PRODEP por el apoyo brindado al proyecto titulado “Conocimiento Semántico en una Interfaz de Lenguaje Natural Portable para Acceder a Información de Bases de Datos Multidimensionales en el Área de Negocios Inteligentes”, con el cual, este artículo fue posible. UACJ-PTC-373.

## Referencias

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American*, 284(5), pp. 34–43 (2001)
2. McGuinness, D.L., Van-Harmelen, F.: OWL Web Ontology Language Overview. W3C recommendation, 10 (2004)
3. Prud'hommeaux, E., Seaborne, A.: SPARQL 1.1 Overview. W3C recommendation 21 (2013)
4. Kaufmann, E., Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users?. In: Franconi, E., Kifer, M., May, W., (ESWC'07), The Semantic Web, Springer, Heidelberg, pp. 281–294 (2007)
5. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: Aroyo, L., Antoniou, G., Hyvonen, E., Ten-Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T., (ESWC'10), Extended Semantic Web Conference, Springer, Heidelberg, 6088, pp. 106–120 (2010)
6. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng: A guided input natural language search engine for querying ontologies. In: Jena User Conference, Citeseer (2006)
7. Tablan, V., Damljanovic, D., Bontcheva, K.: A Natural Language Query Interface to Structured Information. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M., The Semantic Web: Research and Applications, Springer Berlin Heidelberg, pp. 361–375 (2008)
8. Cimiano, P.: Orakel: A natural language interface to an f-logic knowledge base. In: Meiziane, F., Métais, E., Natural Language Processing and Information Systems, Springer Berlin Heidelberg, Heidelberg, pp. 401–406 (2004)



9. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. Gómez-Pérez, A., Euzenat, J., The Semantic Web: Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 546–562 (2005)
10. Damjanovic, D., Agatonovic, M., Cunningham, H.: Identification of the Question Focus: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: 7th Language Resources and Evaluation Conference (LREC), La Valletta (2010)
11. Aduna, J.B., Aduna, A.K.: SeRQL: A Second Generation RDF Query Language. In: SWAD-Europe Workshop on Semantic Web Storage and Retrieval, pp. 13–14 (2003)
12. Kifer, M., Lausen, G.: F-logic: A higher-order language for reasoning about objects, inheritance, and scheme. In: ACM SIGMOD Record, (ACM), 18, pp. 134–146 (1989)
13. Cimiano, P., Haase, P., Heizmann, J.: Porting natural language interfaces between domains an experimental user study with the orakel system. In: Proceedings of the 12th international conference on Intelligent user interfaces, ACM, pp. 180–189 (2008)
14. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), pp. 168–175 (2002)
15. Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A.: WebProtege: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. Semantic Web Journal. IOS Press, 4(1), pp. 89–99 (2013)
16. APACHE Jena: <https://jena.apache.org/> (2017)
17. Zelle, J.M., Mooney, R.J.: Learning to parse database queries using inductive logic programming. In: Proceedings of the 14th National Conference on Artificial Intelligence, pp. 1050–1055 (1996)
18. Padró, L.: Analizadores Multilingües en Freeling. In: Lingüamática, 3, pp. 13–20 (2011)
19. Hunt, J.: Java Server Pages. In: Java and Object Orientation: An Introduction, Springer London, pp. 3361–370 (2002)
20. Vukotic, A., Goodwill, J.: Introduction to Apache Tomcat 7. Apache Tomcat, Apress, Berkeley, 7, pp. 1–15 (2011)
21. Hohensee, B., Hidalgo, I.: Introducción a Android Studio. Incluye Proyectos Reales y El Código Fuente. Babelcube Incorporated (2014)
22. Falkner, J., Jones, K.: Servlets and JavaServer Pages TM The J2EE TM Technology Web Tier. Addison-Wesley (2004)
23. Deitel, P., Deitel, H.: Java Cómo Programar. Pearson Education (2012)
24. Community, A.O.D.: Página de inicio Apache OpenNLP, <https://opennlp.apache.org> (2018)
25. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL), System Demonstrations. pp. 55–60 (2014)
26. Schreiber, G., Amsterdam, V.U., Raimond, I.: BBC: RDF 1.1 Primer. W3C Working Group Note 24 (2014)



# Una representación basada en esquemas preconceptuales de eventos determinísticos y aleatorios tipo señal desde dominios de software científico

Paola Andrea Noreña Cardona, Carlos Mario Zapata Jaramillo

Universidad Nacional de Colombia, Medellín,  
Colombia

{panorenac, cmzapata}@unal.edu.co

**Resumen.** Un evento representa un suceso que dispara un flujo de procesos. Una señal es un tipo de evento que inicia una alerta para cambiar un estado del sistema; estos eventos se clasifican en determinísticos y aleatorios. Determinísticos cuando su efecto se identifica con precisión y aleatorios cuando su causa no se puede predecir. Estos eventos se representan gráfica y matemáticamente en el modelado científico; en ingeniería de software se modelan gráficamente sin componentes científicos y los eventos no se clasifican en determinísticos y aleatorios. Por ello, hace falta un modelo en ingeniería de software que integre componentes científicos para representar eventos determinísticos y aleatorios tipo señal. En este artículo se propone mediante el esquema preconceptual; este esquema es un modelo gráfico de ingeniería de software para representar un dominio. La representación propuesta permite a los analistas representar dominios de software científico que incluyan este tipo de eventos.

**Palabras clave:** dominios de software científico, representación de eventos, evento determinístico, evento aleatorio, señal, esquemas preconceptuales.

## Deterministic/Random Signal-Type Events Representation from Scientific Software Domains based on Pre-conceptual Schemas

**Abstract.** An event represents an occurrence, which triggers a process flow. A signal is an event type starting an alert for changing a state of the system; such events are classified as deterministic and random. Deterministic has an effect precisely identified and random produces an unpredictable cause. Such events are graphically and mathematically represented in scientific modeling; also, they are graphically modeled in software engineering, but such models lack scientific components; also, they lack the deterministic and random event classification. Thus, software engineering lacks a model with scientific components for representing deterministic/random signal-type events. In this paper, we propose a representation about such events from scientific software domains by using pre-conceptual schemas; such schemas are software engineering graphical models for

representing a domain. Such representation is proposed for allowing analysts to represent scientific software domains by including such event types.

**Keywords:** scientific software domain, event representation, deterministic event, random event, signal, pre-conceptual schemas.

## 1. Introducción

Los eventos representan sucesos que, al cumplir con condiciones y restricciones, disparan el flujo de procesos y propician cambios de estado en el comportamiento del sistema [1,2]. Los eventos tipo señal aparecen en el sistema para generar una alerta de un cambio de estado en las variables del sistema [3]. Estos eventos establecen la comunicación entre un emisor que transmite la señal y un receptor que la recibe [4]. Estas señales pueden ser eventos determinísticos o aleatorios; el término *determinístico* permite describir eventos a partir de ecuaciones matemáticas y reglas o condiciones; los eventos determinísticos tipo señal incluyen un valor resultante (efecto) [5]. El término *aleatorio* permite describir un conjunto de datos no determinísticos; los eventos aleatorios tipo señal no se pueden predecir con precisión y, generalmente, se describen mediante variables aleatorias, ya que se desconoce el momento en que van a suceder [6]. Ambos tipos de eventos surgen producto de la ocurrencia de otros eventos [5].

Muñoz *et al.* [7], de las Morenas *et al.* [8] y Ramírez y Colina [9] representan eventos determinísticos tipo señal y Priyadharshini y Justin [10] y Di Leo *et al.* [11] representan eventos aleatorios tipo señal mediante modelos gráficos y matemáticos del modelado científico. En los anteriores enfoques no se agrupa la representación matemática de los eventos en la representación gráfica. Pascual *et al.* [12] modelan matemáticamente eventos aleatorios. Armendáriz [6] propone un enfoque que integra los eventos determinísticos y aleatorios tipo señal en un modelo científico. En el OMG (Object Management Group) [13–14] se propone la representación gráfica de eventos tipo señal en ingeniería de software, sin diferenciar su clasificación científica. Aunque Bazydło *et al.* [15] y Armas *et al.* [16] describen una traducción de modelos de ingeniería de software a modelos científicos, tampoco se enfocan en su clasificación.

Por lo anterior, la ingeniería de software carece de un modelo que integre componentes científicos (gráficos y matemáticos) para representar eventos determinísticos y aleatorios tipo señal desde dominios de software científico (estos dominios involucran las diferentes áreas científicas como ciencias, matemáticas, ingenierías y medicina, entre otros [17]). Adicionalmente, el modelado científico carece de un modelo que integre componentes de ingeniería de software en sistemas que incluyen estos eventos.

En este artículo se propone una representación de eventos determinísticos y aleatorios tipo señal en dominios de software científico a partir de esquemas preconceptuales (EP). Esta representación permite una vista de sistemas que operan con señales como el sistema de telefonía celular. El EP es un modelo de ingeniería de software para representar el dominio de un sistema a partir de componentes dinámicos (que permiten representar la parte comportamental del sistema) y estructurales (que permiten representar las relaciones entre conceptos). Estos componentes le dan una vista completa del dominio a los analistas y a los interesados del sistema [18].

La representación de eventos determinísticos y aleatorios tipo señal en el EP integra componentes con los que se representan estos eventos en el modelado científico. La integración de estos componentes permite que la ingeniería de software tenga un modelo para representar eventos en sistemas de software científico.

Este artículo se estructura de la siguiente manera: en la Sección 2 se presenta el marco teórico; en la Sección 3 se exponen los antecedentes; en la Sección 4 se plantea el problema, en la Sección 5 se propone la solución. Finalmente, se presentan las conclusiones y el trabajo futuro.

## 2. Marco teórico

### 2.1. Eventos

Los eventos representan sucesos o acontecimientos de algo significativo en el flujo de procesos que, al cumplir con condiciones y restricciones, disparan el comienzo o el fin de una operación. Los eventos son responsables del cambio de estados en el comportamiento del sistema; así, la información que se obtiene a partir de los eventos permite analizar este comportamiento. Por ello, la representación de los eventos se vuelve necesaria en la ingeniería de software [1, 2]. El evento que dispara el comienzo de un proceso, un flujo de procesos u otro evento se conoce como *evento disparador* [19]. El evento que surge como resultado del fin de un proceso o un flujo de procesos se conoce como evento de resultado [19, 20].

**Eventos tipo señal:** Estos eventos constituyen eventos disparadores que, mediante una alerta, cambian el estado de las variables del sistema [3]. Según la notación del modelado de procesos de negocio (BPMN, por sus siglas en inglés), los eventos tipo señal se utilizan para representar señales en un sistema de comunicación, donde un transmisor envía una señal y un receptor la recibe [4]. En dominios científicos, como la electrónica, estas señales se clasifican en eventos determinísticos y eventos aleatorios, que pueden surgir producto de otros eventos. De acuerdo con esta clasificación, los analistas científicos pueden modelar estas señales a partir de modelos gráficos y matemáticos [5].

**Evento determinístico tipo señal:** Se representan con ecuaciones matemáticas y reglas o condiciones. Estos eventos implican un valor resultante que es predecible, debido a que la relación causa-efecto se conoce en su totalidad durante la ocurrencia del evento [5].

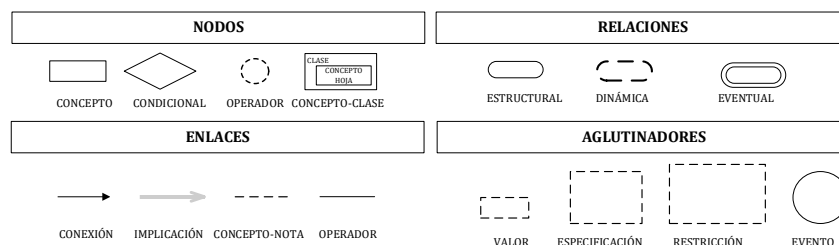


Fig. 1. Elementos de la notación del EP [18].

**Evento aleatorio tipo señal:** Se representa con variables aleatorias o ecuaciones matemáticas con información variable que no puede ser determinística, ya que no se conciben valores predecibles, por lo que se describe un conjunto de valores dentro de los resultados posibles que representan una posible causa [6]. Un evento puede ser aleatorio en su causa y su efecto puede generar un evento determinístico. Por ejemplo, cuando una persona pasa por un sensor en un instante impredecible (evento aleatorio) y genera que la luz se encienda (evento determinístico); ambos eventos requieren el tiempo para identificar su ocurrencia.

## 2.2. Esquema preconceptual (EP)

El EP es un modelo de ingeniería de software para representar computacionalmente un dominio a partir de componentes estructurales y dinámicos. Estos componentes se incluyen para solventar la vista completa del dominio y facilitar su comprensión gráfica. Además, el EP se utiliza como modelo base en UNC-Method (Método de educación de requisitos de la Universidad Nacional de Colombia) para relacionar todos los productos de trabajo en el proceso de ingeniería de software [18].

Los analistas de negocios pueden utilizar la notación gráfica del EP para representar eventos y demás elementos de un dominio mediante herramientas de modelado como Visio de Microsoft™ y Draw.io de Google (gratuita). Los elementos de la notación del EP se basan en reglas lingüísticas (véase la Fig. 1): los *nodos* pueden ser *concepto* (por ejemplo, celular), *condicional* (por ejemplo, si estado de llamada = “conectada”), *operador* (operadores lógicos: *and*, *or* y aritméticos: *\*/*, *+*, *-*) y *concepto-clase* (por ejemplo, “llamada” es concepto clase y “estado” es su concepto hoja); las *relaciones* pueden ser *estructural* (verbo de dependencia *tiene*), *dinámica* (verbo de acción; por ejemplo, persona guarda informe; también se utilizan en procesos automáticos; por ejemplo, actualiza potencia del celular) y *eventual* (verbo de evento, que no requiere un sujeto; por ejemplo, tiempo pasa); los *enlaces* pueden ser *conexión* (enlace entre un concepto y una relación; por ejemplo, alerta→emerge), *implicación* (enlace causa-efecto; por ejemplo, inserta potencia→inserta estación), *concepto-nota* (enlace valor, restricción y especificación) y *operador*; y los *aglutinadores* pueden ser *valor* (por ejemplo, ‘2’), *especificación* (por ejemplo, estado de llamada= “desconectada”), *restricción* (por ejemplo, si potencia de estación >= potencia de celular→inserta potencia del celular) y *evento* (agrupa concepto y verbo de un evento; por ejemplo llamada inicia).

## 2.3. Dominios de software científico

Los dominios de software científico representan el conocimiento de sistemas de software, que incluyen información específica acerca de contextos científicos. Estos dominios contienen información en áreas de ingeniería, medicina, ciencias, matemáticas, administración y economía. Los sistemas de software en estos dominios se utilizan en la resolución de problemas y descripción de fenómenos naturales, biológicos o mecánicos. Estos sistemas se clasifican en *software industrial* (sistemas de software con procesos complejos que expertos científicos apoyan), *software de investigación* (sistemas de software que respaldan una investigación específica en un

dominio científico) y *software a pequeña escala* (sistemas de software que desarrollan investigadores o estudiantes en las diferentes áreas científicas) [17].

### **3. Antecedentes**

En algunos enfoques del modelado científico se incluye la representación de eventos determinísticos tipo señal mediante modelos gráficos y matemáticos. Muñoz *et al.* [7] presentan una red de Petri y un modelo matemático para representar la combinación de señales de entrada y salida. Sin embargo, no incluyen el modelo matemático en la red de Petri. De las Morenas *et al.* [8] implementan un control de un centro de distribución automatizado mediante agentes físicos y dispositivos de señales de radiofrecuencia RFID; ellos representan estas señales mediante redes de Petri, pero no presentan estructuras matemáticas para ellas. Ramírez y Colina [9] incluyen eventos determinísticos tipo señal en una red neuronal para sensores, los cuales verifican la presencia de señales fuera del rango de operación; estas señales indican fallos en la cantidad y calidad de la producción de los pozos de petróleo. La red neuronal se basa en un modelo matemático, pero no se integra a la red.

En otros enfoques del modelado científico se incluye la representación de eventos aleatorios tipo señal mediante modelos gráficos y matemáticos. Priyadharshini y Justin [10] representan los eventos aleatorios tipo señal en un diagrama de bloque; estos eventos son señales de energía en pacientes con válvula mecánica cardíaca y estenosis pulmonar; aunque los eventos también se representan mediante ecuaciones matemáticas, no se integran al diagrama de bloque. Di Leo *et al.* [11] presentan una simulación del modelo matemático en la detección de cambios en señales aleatorias de estructuras turbulentas en un flujo de aire; también, Pascual *et al.* [12] representan eventos aleatorios tipo señal con un modelo matemático en procesos Gaussianos para la varianza del ruido en sensores. Sin embargo, estos enfoques no tienen una representación gráfica de los modelos matemáticos.

Por otra parte, Armendáriz [6] incluye eventos determinísticos y aleatorios tipo señal mediante un autómata finito y modelos matemáticos para detectar fallos, por variación de la frecuencia en la rotación de la hélice de un avión; a pesar de ello, no se integran las estructuras de los modelos matemáticos en el autómata finito. Los anteriores enfoques carecen de componentes de ingeniería de software.

En algunos enfoques del modelado en ingeniería de software se proponen modelos gráficos de eventos tipo señal: En OMG se proponen estructuras gráficas en el modelo de procesos BPMN [13] y en el diagrama de máquina de estados [14] del lenguaje de modelado unificado (UML por sus siglas en inglés); ambos enfoques son modelos tradicionales que sólo permiten presentar una vista dinámica o estructural de un sistema de software, los cuales carecen de la integración de estructuras matemáticas para representar eventos determinísticos y aleatorios. En otros enfoques de este modelado, se describe una traducción de diagramas de ingeniería de software al modelado científico: Bazydlo *et al.* [15] proponen un diagrama de máquina de estados UML que incluye eventos tipo señal, exportan el diagrama al lenguaje de marcado extensible (XML por sus siglas en inglés) y lo traducen al modelo de máquina de estados finitos jerárquicos (HCFSM por sus siglas en inglés) para su posterior simulación; Armas *et al.* [16] plantean una traducción de eventos desde un modelo de procesos BPMN a una

red de Petri. Sin embargo, ninguno de estos enfoques contempla componentes científicos como las estructuras matemáticas para la representación de eventos en la traducción, ni se enfocan en la clasificación científica de los eventos determinísticos y aleatorios.

#### **4. Planteamiento del problema**

Según los enfoques de la Sección 2, usualmente en el modelado científico dos modelos acompañan la representación de eventos determinísticos y aleatorios tipo señal: el modelo gráfico y el modelo matemático; esto permite inferir que ambos modelos se requieren en la representación de estos eventos. A diferencia de este modelado, en ingeniería de software se utilizan sólo modelos gráficos para representar eventos tipo señal y no se enfocan en la clasificación científica de eventos determinísticos y aleatorios, es decir que hacen falta componentes científicos. Además, los modelos tradicionales sólo presentan una vista del sistema (dinámica o estructural). Por ello, la ingeniería de software carece de un modelo que integre componentes científicos (gráficos y matemáticos) para representar eventos determinísticos y aleatorios tipo señal en dominios de software científico. Adicionalmente, se requiere este modelo porque los analistas científicos no integran procesos de ingeniería de software en el desarrollo de software [21,22] y los analistas de negocios (expertos en prácticas de ingeniería de software) no tienen el conocimiento y los componentes para entender estos dominios y producir software científico.

#### **5. Propuesta de solución**

La propuesta de solución se basa en los EP como modelos para la representación de los eventos, ya que en los EP se integran componentes de software y de lingüística computacional a diferencia de los demás modelos como: i) componentes estructurales y dinámicos en un mismo modelo para representar la vista completa del sistema [18]; ii) estructuras lingüísticas, gráficas y matemáticas para representar eventos disparadores [18,20]; esto permite la integración de estructuras matemáticas y demás componentes que proceden de modelos matemáticos para representar eventos determinísticos y aleatorios tipo señal en dominios de software científico.

##### **5.1. Integración de componentes científicos para representar eventos determinísticos y aleatorios tipo señal**

Las ecuaciones matemáticas son “autocontenidas”, es decir, integran elementos de un contexto en la operación; para entender estos elementos se requiere conocer previamente la documentación del contexto en el cual se aplican. Mediante la solución propuesta se pretende integrar las estructuras matemáticas con los elementos del contexto que permitan entender el comportamiento de los eventos determinísticos y aleatorios tipo señal en el mismo modelo a partir de la notación del esquema preconceptual. Para ello, esta fase se divide en dos etapas:



**Traducción de elementos de una ecuación matemática a su forma conceptual:**

Los eventos determinísticos se suelen representar con ecuaciones matemáticas que permiten dar un resultado predecible [24]. Para los eventos tipo señal, las ecuaciones representan el efecto que causa la señal; por ejemplo, al observar la ecuación (1): i) se debe indicar el contexto en el cual se fórmula, es decir, la ecuación de la potencia que recibe un celular desde una estación transmisora en la red de telefonía celular y ii) se deben identificar los elementos que se encuentran autocontenidos, es decir,  $Pr$  es la potencia de señal recibida,  $Pt$  es la potencia transmitida de la estación,  $cte$  es la constante de propagación de la señal,  $d$  es la distancia entre la estación (transmisor) y el celular (receptor) y  $n$  es el coeficiente de propagación del entorno, que en valores urbanos puede tomar valores entre 2,5 y 5 W (Watts) [25]:

$$Pr = \frac{Pt}{cte} * d^n. \quad (1)$$

Para traducir esta ecuación al evento tipo señal *señal aparece* en el celular, se utiliza una regla de los EPs que exige que los elementos que se describen en su notación deben ser conceptos con palabras completas que faciliten la comprensión del dominio. De esta manera, se utiliza el nombre de cada variable y parámetro (constante) de la ecuación. Al tomar el mismo ejemplo, la ecuación (1) se presenta en su forma conceptual en la ecuación (2).

$$Potencia\ recibida = \frac{Potencia\ transmitida}{Constante\ de\ transmisión} * distancia^{coeficiente\ de\ propagación\ del\ entorno}. \quad (2)$$

Los eventos aleatorios se suelen representar mediante ecuaciones con variables aleatorias que dan una probabilidad al valor. Estos valores también se utilizan para efectuar simulación de sistemas. Para los eventos tipo señal se relaciona al momento impredecible en que ocurre la señal. La función *random* en las plataformas de desarrollo de software se integran para generar valores aleatorios entre 0 y 1 y generalmente se presenta como *rand()*. Para aplicar esta función se asigna a la variable aleatoria y se traduce como se presenta en la ecuación (3). Esta función puede variar según el requisito. Si se requiere que el valor aleatorio se encuentre entre dos valores, se utiliza la ecuación (4), donde  $X$  es el valor aleatorio,  $m$  es el límite inferior y  $n$  es el límite superior. De esta manera, se traduce en la ecuación (5); si el requisito es para dos valores, se utiliza la función *rand* con ambos valores numéricos o valores *string* (caracteres):

$$Variable\ aleatoria = rand, \quad (3)$$

$$X = m + rand() \% n + 1 - m, \quad (4)$$

$$Variable\ aleatoria = límite\ inferior + rand \% límite\ superior + 1 - límite\ inferior. \quad (5)$$

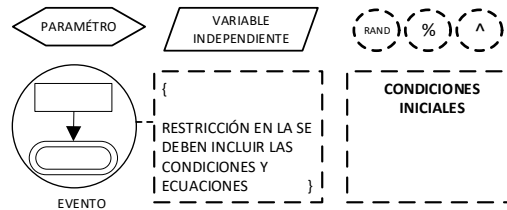


Fig. 2. Estructuras para representar eventos determinísticos y aleatorios tipo señal.

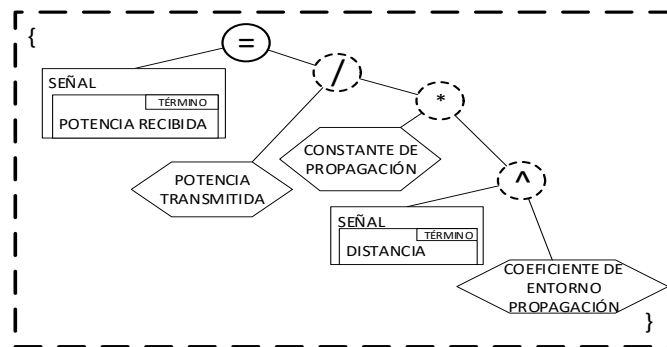


Fig. 3. Representación de la ecuación de la potencia transmitida.

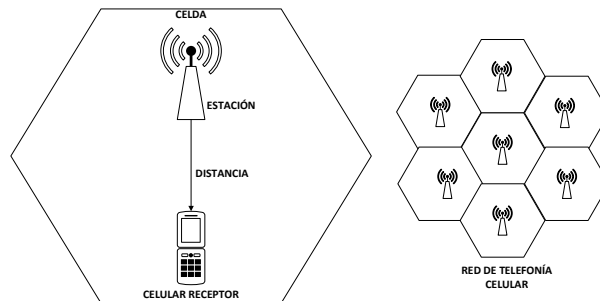


Fig. 4. Sistema de telefonía celular a partir del celular receptor, elaboración propia a partir de [25].

**Definición de estructuras para representar eventos determinísticos y aleatorios tipo señal:** Se aplica la notación de las restricciones del EP para definir estructuras que permitan la representación de eventos determinísticos y aleatorios tipo señal, como condiciones y ecuaciones matemáticas desde dominios de software científico a partir de la traducción en la etapa anterior. Para las ecuaciones matemáticas se utilizan elementos gráficos de la notación del esquema, como conceptos, operadores y conectores de operadores, y se integran otras estructuras como el *parámetro*, la *variable independiente* para definir variables globales como el tiempo; algunos operadores como el operador *rand*, el *porcentaje* (%) para generar variables aleatorias y el *operador exponencial* (^) con base en un valor y las *condiciones iniciales* que se utilizan en dominios de software científico para simular los sistemas (véase la Fig. 2).

Las ecuaciones matemáticas se elaboran a partir de árboles binarios, que comúnmente se usan para representar expresiones algebraicas y booleanas [23]. Por ejemplo, para representar la ecuación (2), se incorporan las estructuras anteriores en una *restricción* que se relaciona al evento (véase Fig. 2) surgiendo la representación de la Fig. 3. Estas restricciones también se utilizan para condiciones del evento.

## 5.2. Una representación basada en esquemas preconceptuales de eventos determinísticos tipo señal desde dominios de software científico

La integración de componentes científicos en el EP se aplica al dominio científico de electrónica en un sistema de red de telefonía celular que, junto con los componentes de software del EP permiten una vista dinámica y estructural del sistema. El nivel de detalle se centra en el funcionamiento del sistema para el celular que recibe una llamada; así, la representación de eventos se orienta a eventos determinísticos y aleatorios tipo señal, que suceden en el celular receptor y las características que se modelan se basan en su ocurrencia respecto del tiempo.

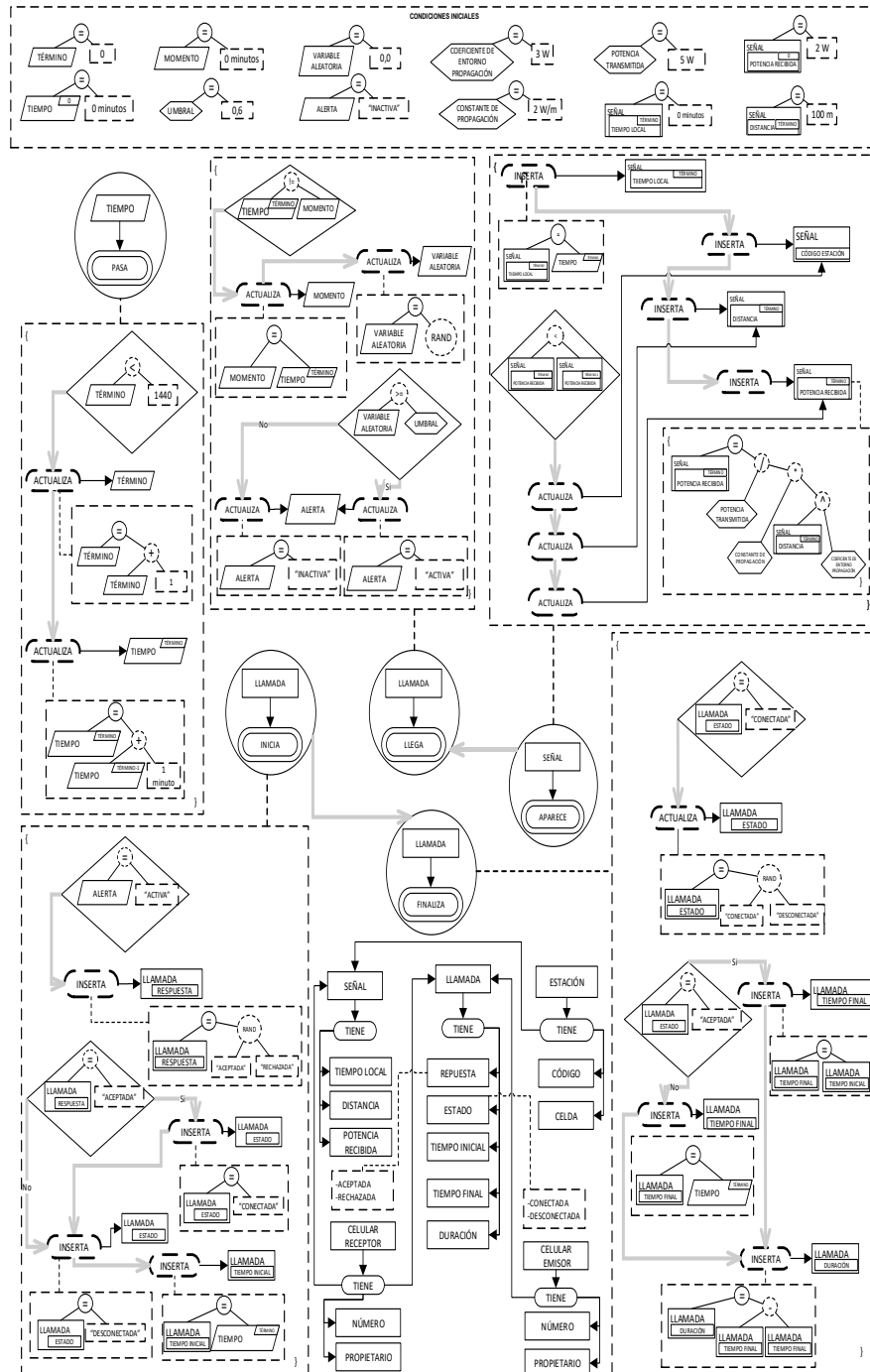
En la Fig. 4 se puede apreciar una red de telefonía celular que se distribuye en celdas, las cuales constituyen el área de cobertura de una estación o antena. Estas estaciones funcionan como transmisores de potencia, que son las que permiten que un celular reciba la potencia de la señal, es así como un celular que hace parte de la red puede recibir una llamada. La señal de los celulares debe ser baja para que otros móviles pueda reutilizar los canales de las celdas [25].

En la Fig. 5 se elabora un EP del dominio del sistema utilizando las estructuras propuestas.

El EP se representa con la funcionalidad automática que integran los simuladores de sistemas de eventos. Este modelo se logró a partir del conocimiento de un experto y fuentes en el área.

Los componentes científicos se aprecian en el EP comenzando en las *condiciones iniciales*, las cuales permiten ejecutar la vista dinámica del sistema con valores iniciales de parámetros y variables globales e internas. La secuencia de eventos inicia con el evento *tiempo pasa*, el cual se representa con una restricción que incluye un ciclo que inicia en '0' según la condición inicial *término=0*, y termina en '1440' según la condición *término < 1440*. *Término* es el contador de vectores que permite controlar la posición del tiempo y el registro en las ocurrencias de cada evento; el valor 1440 identifica las iteraciones por el *tiempo 1440 minutos* que equivalen a un 1 día, el cual incrementa de por cada '1 minuto'. Este evento controla la secuencia de los demás eventos del sistema, ya que cada ocurrencia se toma respecto del tiempo.

El evento *señal aparece* representa la señal que el *celular receptor* tiene a partir de la *estación* transmisora. Este evento tipo señal es determinístico, ya que se puede representar mediante la condición si *potencia recibida[término]* de la *señal* < *potencia recibida[término-1]* (que permite al celular buscar la señal más alta de las estaciones de transmisión) y la ecuación (2), dando un resultado predecible de la potencia recibida, que se deriva de los elementos de la fórmula y las condiciones iniciales *potencia transmitida=5 W* (valor por normatividad que deben emitir las estaciones transmisoras y que puede variar según la regulación de un país), *la constante de propagación=2 W/m* y el *coeficiente de entorno propagación=3 W* (puede variar según el entorno urbano o rural, ya que el entorno urbano requiere más celdas en el sistema).



**Fig. 5.** EP de eventos determinísticos y aleatorios tipo señal en el sistema de telefonía celular desde el funcionamiento del celular receptor.

La especificación de este evento contiene procesos automáticos que se ejecutan para guardar el valor de la *potencia recibida*; de esta manera, el celular constantemente guarda el registro en un instante de tiempo, como se observa en la relación dinámica *inserta tiempo local[término]*, luego *inserta el código* de la *estación* a la que se está conectando el celular receptor y la *distancia* entre esta estación y el celular. La *distancia* hace que la señal varíe, por lo cual, si es menor que la anterior se *actualiza código estación*, se *actualiza distancia[término]*, para finalmente *actualizar potencia recibida[término]*.

**Tabla 1.** Funcionamiento del evento *tiempo pasa*.

TIEMPO PASA	
TÉRMINO	TIEMPO
0	0 minutos
1	1 minuto
120	120 minutos
1440	1440 minutos

**Tabla 2.** Funcionamiento del evento *llamada llega*.

LLAMADA LLEGA		
MOMENTO	VARIABLE ALEATORIA	ALARMA
0 minutos	0,0	INACTIVA
1 minuto	0,6	INACTIVA
260 minutos	0,7	ACTIVA
1440 minutos	0,8	ACTIVA

**Tabla 3.** Tabla del concepto-clase *celular receptor*.

CELULAR RECEPTOR	
NÚMERO	PROPIETARIO
3002342111	Paola Noreña

La *llamada llega* es un evento aleatorio, el cual se representa a partir de una condición que guarda el instante (impredecible) en el que puede ocurrir el evento; si esto ocurre se activa la llamada. La *llamada llega* no implica que se inicie una llamada, ya que el receptor puede aceptar o rechazar la llamada, el evento *llamada inicia* surge a partir de la respuesta del receptor; sin embargo, no es de interés para los expertos del dominio de electrónica conocer la persona o rol que ejecuta la acción, sino que se enfoca en el funcionamiento de los elementos del sistema. Es así como, en la representación de este evento, se incluye la respuesta aleatoria, que puede ser “*aceptada*” o “*rechazada*”, lo que requiere conocer la comunicación que se establece en el sistema al ser “*aceptada*”, inserta el *estado* de la llamada “*conectada*” y el *tiempo inicial*; luego surge el evento *llamada finaliza* a partir del *estado* de la llamada “*desconectada*”, con el cual se inserta *tiempo final* y *duración*.

La vista estructural del sistema en el EP, permite representar la información que se deriva del sistema a partir de los eventos y los conceptos clase son *estación* (cuyos

**Tabla 4.** Tabla del concepto-clase *celular emisor*.

CELULAR EMISOR	
NÚMERO	PROPIETARIO
3203234556	Luis Fernando Guarín
3142342112	Martha Lucía Cardona

**Tabla 5.** Tabla del concepto-clase *estación*.

ESTACIÓN	
CÓDIGO	CELDA
123.12.21.1	Robledo
123.12.21.3	Laureles

**Tabla 6.** Tabla del funcionamiento del evento *señal aparece*.

SEÑAL APARECE				
TÉRMINO	TIEMPO LOCAL	CÓDIGO DE ESTACIÓN	DISTANCIA	POTENCIA RECIBIDA
0	0 minutos	120.34.23.1	100 m	2 W
1	1 minuto	120.34.23.1	1000 m	3 W
2	2 minutos	120.34.23.1	150 m	4 W

**Tabla 7.** Tabla del concepto-clase *llamada*.

LLAMADA						
NUMERO CELULAR EMISOR	NÚMERO CELULAR RECEPTOR	RESPUESTA	ESTADO	TIEMPO INICIAL	TIEMPO FINAL	DURACIÓN
3203234556	3002342111	Aceptada	Desconectada	120 minutos	180 minutos	60 minutos
3142342112	3002342111	Rechazada	Desconectada	220 minutos	220 minutos	0 minutos
3142342112	3002342111	Aceptada	Desconectada	430 minutos	460 minutos	30 minutos

conceptos hoja o atributos son *celda*, *ubicación* y *potencia*), *celular receptor* (cuyos conceptos hoja son *número* y *propietario*), *celular emisor* (cuyos conceptos hoja son *número* y *propietario*), *estación* (cuyos conceptos hoja son *código* y *celda*), *señal* (cuyos conceptos hoja son *tiempo local*, *distancia* y *potencia recibida*) y *llamada* (cuyos conceptos hoja son *respuesta—aceptada o rechazada*, *estado* (*conectada*, *desconectada*), *tiempo inicial*, *tiempo final* y *duración*).

Mediante las siguientes tablas de datos se puede verificar el funcionamiento de la vista estructural del EP y de la base de datos del sistema de software.

## 6. Conclusiones y trabajo futuro

La representación que se propuso en este artículo basada en el EP permite modelar eventos determinísticos y aleatorios tipo señal en dominios de software científico, en una vista completa del dominio que integra la vista estructural y la vista dinámica. Esta representación incluye elementos nuevos al EP, como las condiciones iniciales, los

parámetros y las variables independientes, además de una serie de operadores comunes a la notación matemática y que previamente no se incluían en el EP. La representación de estos eventos en el EP constituye un enfoque que permite unificar componentes científicos y componentes de software en un mismo modelo de ingeniería de software para la producción de software científico.

Analistas científicos y analistas de negocios pueden utilizar el EP para representar y comprender el funcionamiento de eventos determinísticos y aleatorios tipo señal en dominios de software científico.

Como trabajo futuro se pueden adicionar otras estructuras complejas para representar otro tipo de eventos y elementos de los dominios de software científico, como ecuaciones diferenciales o análisis estadísticos.

**Agradecimientos.** Este artículo es producto del proyecto de investigación Doctoral *una extensión a esquemas preconceptuales para el refinamiento en la representación de eventos y la notación matemática* con código Hermes 39886 de la Universidad Nacional de Colombia, que financia Colciencias en la convocatoria 727 de becas para estudiantes de Doctorado en Colombia.

## Referencias

1. Zapata, C.M., Noreña, P.A., Gonzales, N.: Representación de eventos disparadores y de resultado en el grafo de interacción de eventos. Ing. UsbMed, 4(2), pp. 23–32 (2013)
2. Noreña, P.A., Zapata, C.M.: A Game for Learning Event-Driven Architecture: Pre-conceptual-Schema-based Pedagogical Strategy. Development in Business Simulation and Experiential Learning, 45, pp. 312–319 (2018)
3. Noreña, P.A., Vargas, F.A., Soto, D.E.: Tipificación de eventos a partir del modelo BPMN en artefactos de ingeniería de software. Cuaderno Activa, 6, pp. 49–61 (2014)
4. Von-Rosing, M., Von-Scheel, H., Scheer, A.W.: The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM. Morgan Kaufmann, 1 (2014)
5. Ruhm, K.H.: Deterministic, Nondeterministic Signals. Internet Portal Measurement Science and Technology. <http://www.mmm.ethz.ch/dok01/d0000839.pdf> (2008)
6. Armendariz, I.: Análisis paramétrico determinista y aleatorio de problemas dinámicos en estructuras aeroespaciales. Tesis Doctoral en Aeronáutica, Universidad Politécnica de Madrid (2016)
7. Muñoz, D.M., Correcher, A., García, E., Morant, F.: Generación determinística de lenguajes legales para sistemas de eventos discretos. Revista Iberoamericana de Automática e Informática Industrial, (RIAI), 13(2), pp. 207–219 (2016)
8. De las Morenas, J., García, A., Martínez, F., Ansola, P.G.: Implementación del control en planta de un centro de distribución automatizado mediante agentes físicos y RFID. Revista Iberoamericana de Automática e Informática Industrial, (RIAI), 12(1), pp. 25–35 (2015)
9. Ramírez, M., Colina, E.: Sistema supervisor inteligente para procesos de producción de petróleo. Revista científica Maskana, en Congreso (MATCH'14), pp. 71–82 (2016)
10. Priyadharshini, V., Justin, J.: Quantification of Non-deterministic Events in Pathological PCG signals using Continuous and Packet Wavelet Transforms. In: International Conference on Communications and Signal Processing (ICCSP'14), pp. 1419–1423 (2014)
11. Di Leo, J.M., Calandra, M.V., Delnero, J.S.: Algoritmos de punto de cambio aplicados a la detección de estructuras vorticosas en flujos turbulentos. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, 33(3), pp. 225–234 (2017)

12. Pascual, J.P., von-Ellenrieder, N., Muravchik, C.: Cramér-Rao Bound for Parameter Estimation in Sensor Arrays with Mutual Coupling. *IEEE Latin America Transactions*, 11(1), pp. 91–96 (2013)
13. OMG, Object Management Group: Business Process Model and Notation BPMN. <http://www.omg.org/spec/BPMN/1.2> (2014)
14. OMG, Object Management Group: Superstructure 2.5. <http://www.omg.org/spec/UML/2.5/> (2015)
15. Bazydło, G., Adamski, M., Stefanowicz, L.: Translation UML diagrams into Verilog. In: 7th International Conference on Human System Interactions (HIS'14), pp. 267–271 (2014)
16. Armas, A., Baldan, P., Dumas, M., Garcia, L.: Diagnosing behavioral differences between business process models: An approach based on event structures. *Information Systems*, 56, pp. 304–325 (2016)
17. Kelly, D.: Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software. *Journal of Systems and Software*, 109, pp. 50–61 (2015)
18. Zapata, C.M.: The UNC-Method revisited: elements of the new approach. Saarbrücken: Lambert (2012)
19. Zapata, C.M., Noreña, P.A., Vargas, F.A.: The Event Interaction Game: Understanding Events in the Software Development Context. *Developments for Business Simulation and Experiential Learning*, 41, pp. 256–263 (2014)
20. Noreña, P.A.: Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method. Tesis de Maestría, Universidad Nacional de Colombia (2013)
21. Wilson, G., Aruliah, D.A., Brown, C.T., Chue-Hong, N.P., Davis, M., Guy, T., Steven-Haddock, H.D., Huff, D., Mitchell, I.M., Plumbley, M.D., Waugh, B., White, E.P., Wilson P.: Best Practices for Scientific Computing. *PLOS Biology*, 12(1), pp. 1–6 (2014)
22. Kanewala, U., Bieman, J.M.: Testing scientific software: A systematic literature review Author links open overlay. *Information and Software Technology*, 56(10), pp. 1219–1232 (2014)
23. Leon, A.: Probability, Statistics, and Random Processes for Electrical Engineering. Pearson Prentice Hall (2017)
24. Kuipers, J., Ueda, T., Vermaseren, J.A.M.: Code optimization in FORM. *Computer Physics Communications*, 189, pp. 1–19 (2015)
25. Cruz-Ornetta, V.: Telefonía móvil y su salud. Lima: INICTEL (2005)



# Un método para el análisis de sentimientos bajo un enfoque supervisado usando recursos léxicos

Antonio Hernández Ambrocio, Gabriela Ramírez de la Rosa,  
Esaú Villatoro Tello

Universidad Autónoma Metropolitana (UAM), Unidad Cuajimalpa,  
Departamento de Tecnologías de la Información,  
México

antonio.hdza12@gmail.com,  
{gramirez,evillatoro}@correo.cua.uam.mx

**Resumen.** La tarea de análisis de sentimientos ha sido un tema de interés desde hace algunos años. Conocer si un texto es positivo o negativo es relevante para tener una opinión general sobre diversas entidades como: productos, personajes públicos, y empresas. En este artículo presentamos un método de clasificación supervisada que toma ventaja de recursos léxicos existentes en español e inglés. Los experimentos realizados están orientados a contestar en qué medida la información externa de recursos léxicos, es útil bajo un enfoque de clasificación supervisada. Los resultados obtenidos muestran que el método propuesto es estable para ambos idiomas y diferentes recursos léxicos empleados.

**Palabras clave:** recursos léxicos, aprendizaje supervisado, análisis de sentimientos, procesamiento del lenguaje natural.

## A Sentiment Analysis Method using a Supervised Approach based on Lexical Resources

**Abstract.** Sentiment analysis task has been an interest topic in recent years. Knowing if a text is positive or negative is relevant to have a general opinion about different entities such as products, public personalities or companies. In this paper we present a supervised classification method that takes into account existing Spanish or English lexicons. The experiments carried out are aimed at answering how much the external information of lexical resources is useful in a supervised classification approach for sentiment analysis. The obtained results show that the proposed method is stable for both languages and for different lexicons used.

**Keywords:** lexical resources, supervised learning, sentiment analysis, natural language processing.

## 1. Introducción

A diario, usuarios activos en la Web generan inmensas cantidades de información en forma de opiniones sobre diversos temas. Estas opiniones son vertidas en medios digitales; por ejemplo en: foros de opinión, blogs, redes sociales, sitios de reseñas o evaluación de productos, sitios de comercio electrónico, entre otros. Un panorama de esta constante actividad de millones de personas al rededor del mundo en redes sociales se puede ver en estadísticas de Facebook y Twitter. Por un lado, Facebook tiene hasta 1.2 billones de usuarios activos al día y 1.9 billones al mes<sup>1</sup>. Por otro lado Twitter reporta un promedio de 500 millones de tweets escritos al día<sup>2</sup>. Así mismo, el sitio de comercio electrónico Amazon cuenta con 2.4 billones de visitas mensuales<sup>3</sup>.

Ante esta creciente cantidad de información es importante contar con métodos automáticos que nos permitan realizar un análisis rápido y eficiente de la información para la toma de decisiones. El área encargada del análisis de este tipo de textos es el Procesamiento del Lenguaje Natural, una sub-área de la Inteligencia Artificial. Así, la tarea del análisis de sentimientos se puede definir como una tarea de clasificación donde a cada texto se le puede asignar una de tres posibles etiquetas: negativa, positiva, y en algunos casos la etiqueta de opinión neutral.

Existen dos principales enfoques de solución para la tarea del análisis de sentimientos: i) enfoques basados en recursos léxicos y ii) enfoques basados en aprendizaje supervisado. En el primer enfoque se necesita de un diccionario de palabras asociadas a un sentimiento (positivo o negativo). Este tipo de diccionarios son compilaciones que capturan conocimiento previo de las palabras que en él aparecen. En el segundo enfoque, aprendizaje supervisado, no se requieren recursos léxicos pero sí un conjunto de ejemplos de opiniones previamente etiquetados.

Obtener ejemplos etiquetados es una tarea costosa, pues usualmente estos ejemplos se etiquetan de forma manual; adicionalmente, es preferible contar con ejemplos específicos para el dominio de clasificación que se necesite. Por ejemplo, si se requiere saber el sentimiento de revisiones de libros, posiblemente los ejemplos etiquetados en este dominio no sean adecuados para determinar el sentimiento hacia un personaje político.

Por otro lado, actualmente se cuentan con diversos recursos léxicos que contienen palabras asociadas a un valor de sentimiento u opinión. Estos recursos pueden ser útiles para clasificar nuevos textos sin importar el dominio. Sin embargo, usar solamente la información de estos recursos léxicos puede no ser adecuado para todos los dominios pues normalmente cada uno tienen formas específicas de expresar opiniones positivas o negativas.

Dado lo anterior, en este artículo se trata de responder a la siguiente pregunta de investigación: ¿en qué medida se puede combinar información de los recursos

<sup>1</sup> All Facebook Statistics In One Place. <https://www.socialbakers.com/statistics/facebook/>

<sup>2</sup> All Twitter Statistics In One Place. <https://www.socialbakers.com/statistics/twitter/>

<sup>3</sup> Web Visitor traffic to Amazon.com. <https://www.statista.com/statistics/623566/web-visits-to-amazoncom/>

léxicos previamente contruidos bajo un enfoque de clasificación supervisado en la tarea de análisis de sentimientos?

El resto de este artículo está organizado como sigue: en la sección 2 se hace un repaso de trabajos pertinentes al tema de análisis de sentimientos. En la sección 3 se presenta la idea general del método propuesto. Luego, en la sección 4 se describen los recursos léxicos usados, y una propuesta de estandarización. Los experimentos realizados se presentan en la sección 5 y finalmente en la sección 6 se exponen ideas de trabajo futuro y conclusiones.

## 2. Trabajo relacionado

Se han realizado numerosos trabajos empleando métodos basados en recursos léxicos, algunos con buenos resultados. Tal es el caso del método que reportó Ding et al. [7] en el 2008, en el que proponen un léxico conformado por *palabras de opinión*. En el método propuesto en ese trabajo se trata de determinar el contexto y reglas semánticas de la oración.

Por otro lado, bajo un enfoque de aprendizaje supervisado, Pang B. et al.[9] experimentaron con tres algoritmos de aprendizaje: Naive Bayes, *Maximum Entropy* y *Support Vector Machines*. Sus resultados sugieren que se puede notar que los clasificadores tienen, en general, una exactitud aceptable.

Adicionalmente, se han propuesto métodos que involucran recursos léxicos y métodos de aprendizaje supervisado. Por ejemplo, Madhavi D. et al.[6] trabajaron con dos enfoques: el primero es la construcción de clasificadores con ensambles que obtienen de la combinación de tres léxicos de cuatro que usan (SentiWordNet, Bing Liu Lexicon, SenticNet, MPQA), haciendo un clasificador por cada léxico, además uno por cada ensamble; el segundo enfoque mencionado consiste en aunar el conocimiento que proporcionan los recursos léxicos a un clasificador. Reportan no haber encontrado una mejora en usar ensambles a usar los léxicos originales, sin embargo, se notó una mejoría al hacer uso del conocimiento proporcionado por los léxicos.

## 3. Método propuesto

En la figura 1 se presenta el esquema general del método de clasificación propuesto. De manera similar a un enfoque de clasificación supervisada, tiene dos etapas: entrenamiento y validación.

**Etapla 1. Entrenamiento usando recursos léxicos.** En esta etapa se construye un modelo de clasificación mediante un conjunto de documentos previamente etiquetados. De la misma forma que en el enfoque tradicional, la representación de dichos textos se hace considerando únicamente información de los documentos etiquetados, es decir, una representación vectorial donde cada vector corresponde a un documento en el corpus etiquetado  $D$ , de tal forma que cada  $d_i \in D$  se define con un vector  $\langle w_{ij}, \dots, w_{im} \rangle$ , usualmente la dimensión

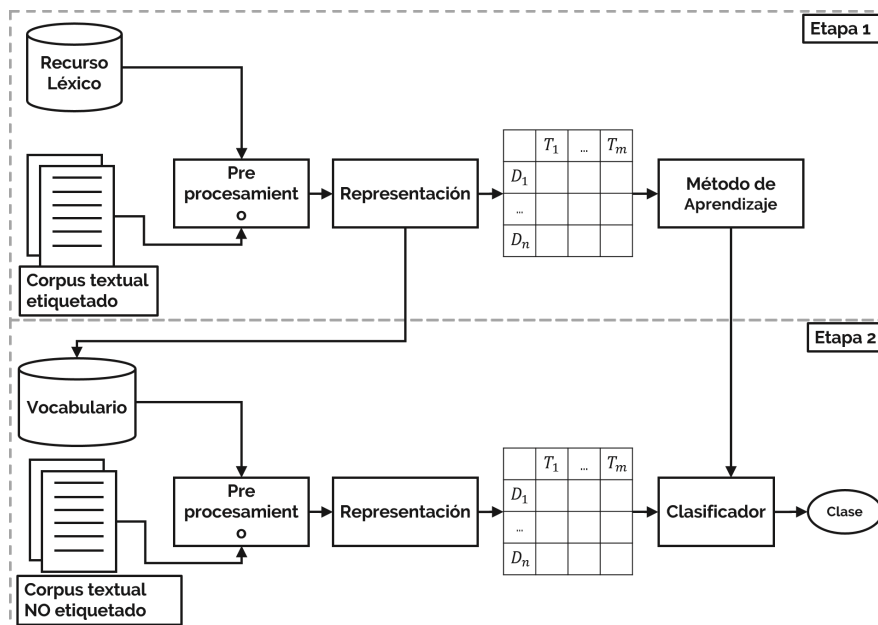


Fig. 1. Esquema general del método propuesto.

del vector corresponde al tamaño de los términos diferentes en el conjunto de documento  $D$ . En nuestra propuesta, la dimensión del vector corresponde al conjunto de términos  $T$  donde  $\forall t \in T \mid t \in (V \cap L)$  siendo  $V$  el vocabulario de todos los documentos en  $D$  y  $L$  el conjunto de términos en un recurso léxico dado. Por lo tanto, una salida parcial de esta etapa es un vocabulario compuesto por el conjunto  $T$ . Note que en nuestra propuesta se incorpora una comparación adicional, que es que el término a considerar en la representación también esté en el conjunto de términos del recurso léxico.

Una vez que se tiene la representación vectorial de cada documento etiquetado se utiliza algún algoritmo de aprendizaje, como Naïve Bayes o SVM, para generar el modelo de aprendizaje específico para esta tarea.

**Etapa 2. Validación.** El propósito de la etapa de validación es usar el mismo conjunto  $T$  de términos para representar los documentos a clasificar, luego se utiliza el modelo de aprendizaje construido en la etapa anterior para determinar la clase de cada documento. Cabe mencionar que este proceso es similar al del enfoque tradicional.

#### 4. Recursos léxicos para el análisis de sentimientos

Un recurso léxico es un conjunto de términos en la que cada uno de estos términos está asociado a un sentimiento (positivo o negativo). El sentimiento

puede ser representado con la polaridad, i.e., positivo; o con un valor numérico que refleja la probabilidad afectiva.

En este artículo utilizamos un conjunto de recursos léxicos tanto para el idioma Inglés como para el Español que han sido usados previamente en la literatura [6,10]. En la tabla 1 se muestran cinco recursos léxicos con el número de palabras asociada a cada sentimiento. Los recursos léxicos que se analizaron y posteriormente usaron en la evaluación experimental de la sección 5 son: SentiWordNet [1], SenticNet<sup>4</sup> y ML-Senticon [5] para el idioma inglés. Además, para el español se usan: la versión en español de SentiCon [5] y una adaptación de ANEW [3] (Por sus siglas en inglés Affective Norms for English Words) en español.

**Tabla 1.** Número de palabras positivas y negativas en cinco recursos léxicos: SentiWordNet, SenticNet, ML-SentiCon, ANEW y MS-SentiCon (versión en español).

Recurso léxico	Idioma	Términos positivos	Términos negativos
SentiWordNet	EN	25,508	26,440
SenticNet	EN	27,405	22,595
ML-SentiCon	EN	12,999	12,424
ANEW	ES	511	523
ML-SentiCon	ES	5,568	5,974

La mayoría de los recursos léxicos mostrados en la tabla 1 cuentan con información similar. Sin embargo, cada recurso léxico puede presentar su lista de términos en diferentes formatos, diferente orden, o algunos tienen información adicional como su etiqueta POS (part-of-speech). Con el propósito de homogeneizar estos recursos se presenta una propuesta de estandarización a un formato compuesto por los pares término - polaridad. Esta polaridad se representa con un valor numérico en un rango de -1 a 1, siendo -1 polaridad negativa y 1 positiva. La intención de esta homogeneización es que el método propuesto pueda ser adaptado para cualquier léxico con este formato.

## 5. Evaluación experimental

En esta sección, mediante una evaluación empírica, se busca evaluar el uso de las palabras de recursos léxicos en el enfoque supervisado de clasificación de sentimientos.

Para todos los experimentos se utilizó la representación vectorial descrita en la sección 3 usando tres diferentes esquemas de pesado: a) booleano, asignando 1 cuando el término  $t_k$  aparece en el documento  $d_i$  y utilizando 0 en otro caso; b) TF (frecuencia del término), asignando el número de veces que el término  $t_k$  aparece en el documento  $d_i$ ; y TF-IDF (frecuencia de término - frecuencia inversa del documento),  $TF(t_k) \times IDF(t_k)$ , donde  $IDF(t_k) = \log \frac{|D|}{|\{d_i \in D: t_k \in d_i\}|}$ .

<sup>4</sup> Disponible en: <http://sentic.net>

Para medir el desempeño de cada configuración se usaron las medidas precisión, recuerdo y la medida F. La precisión ( $P$ ) es la proporción de instancias clasificadas correctamente en una clase  $c_i$  con respecto a la cantidad de instancias clasificadas en esa misma clase. El recuerdo ( $R$ ), la proporción de instancias clasificadas correctamente en una clase  $c_i$  con respecto a la cantidad de instancias que realmente pertenecen a esa clase. Usando estas medidas es posible tener evaluación global del sistema de clasificación mediante el cálculo de la medida F, definida como lo indica la Ecuación 1:

$$F = \frac{(1 + \beta^2)P * R}{\beta^2(P + R)}. \quad (1)$$

Adicionalmente, los algoritmos de clasificación usados en los experimentos fueron *Naive Bayes* y el algoritmo *Optimización Mínima Secuencial* (SMO) con la implementación de Weka [8] usando los parámetros por defecto. Finalmente, para todos los experimentos se usó un esquema de validación cruzada a 10 pliegues

En las siguientes sub-secciones se describen las colecciones de datos usados, el pre-procesamiento realizado a estas colecciones de datos y se describen tres experimentos: i) evaluación del uso de los recursos léxicos en el esquema descrito previamente; ii) evaluación de ensambles de recursos léxicos; y iii) evaluación de la utilización del valor afectivo asociado a cada término en algún recurso léxico.

### 5.1. Colección de datos

Para los experimentos se usaron colecciones de datos en español e inglés. El corpus utilizado para el idioma inglés fue construido por John Blitzer, et al [2] y está compuesto por opiniones vertidas en Amazon de cuatro tipos de productos (dominios): Libros, DVDs, Electrónicos, y Artículos de Hogar. Cada dominio contiene una colección de 2000 opiniones, 1000 por cada clase. En este corpus, las opiniones están asociadas a una puntuación de 0 a 5 estrellas al producto. A partir de esta puntuación se etiquetaron las opiniones en dos clases: las opiniones con una calificación mayor a 3 fueron etiquetadas como positivas y las opiniones con calificación menor a 3 se les etiquetó como negativas.

Por otro lado, el corpus en español cuenta con un total de 3,553 críticas de cine extraídas del sitio *www.muchocine.net* [4]. Cada crítica tiene asociada una calificación que va de 1 a 5. Para separar las críticas, los autores, consideraron como positivas a todas aquellas que contaran con una calificación mayor a 3, y en caso contrario a todas las críticas que tuviesen una calificación menor de 3, se les consideró negativas, el resto fueron ignoradas. Tras la separación de las críticas, según su calificación, el corpus fue reducido a 2,277 críticas, de las cuales 1,147 resultaron ser positivas, y 1130 negativas.

### 5.2. Pre-procesamiento

Antes de realizar los experimentos se requirió un pre-procesamiento de los documentos de ambas colecciones de datos. De manera general, se eliminaron

signos de puntuación, se convirtieron todas las letras a minúscula, y se hizo un parseo a codificación UTF-8 para evitar pérdida de información.

Dado que los términos utilizados para la representación de los documentos deben estar en un recurso léxico, no se requirió hacer otro tipo de pre-procesamiento a estos documentos.

### 5.3. Evaluando recursos léxicos en un enfoque supervisado

En este primer experimento se busca determinar el desempeño del modelo de aprendizaje usando una representación basada únicamente en un recurso léxico (de los descritos en la sección 4). Las tablas 2 y 3 muestran los resultados obtenidos para las colecciones en inglés y español, respectivamente.

De la tabla 2 podemos observar que los mejores resultados para tres dominios se obtienen cuando se usa el recurso léxico SentiWordNet usando el algoritmo de aprendizaje SMO. Para el dominio Libros, la representación que usa el léxico SenticNet obtuvo el mejor resultado. Detrás se encuentra SenticNet y por último ML-SentiCon. En todos los casos sobresalió el uso del pesado booleano en la representación.

Para la colección en español, se usaron los recursos léxicos ANEW y ML-SentiCon, los resultados en medida F se muestran en la tabla 3. Respecto a los resultados obtenidos de los recursos en español, el léxico SentiCon fue con el que se obtuvieron los mejores resultados. Note que el clasificador que funcionó mejor fue el construido bajo el algoritmo SMO con pesado booleano con un medida F de 0.707.

De forma general, de este primer experimento podemos concluir que el tamaño del recurso léxico no importa de forma relevante en el desempeño de clasificación. Por ejemplo, para el caso del corpus en español, la diferencia de la cantidad de palabras que contiene ANEW contra ML-SentiCon es 10 veces menor (ver tabla 1) pero el mejor resultado obtenido con ANEW es 0.68 de medida F mientras que con ML-SentiCon es 0.707 de medida F.

### 5.4. Evaluando la complementariedad de los recursos léxicos

La idea detrás de este segundo experimento es determinar si combinando información proveniente de los recursos léxicos disponibles es posible mejorar el desempeño de clasificación. Para probar la idea se realizaron dos *ensambles* de recursos léxicos. El primer ensamble, llamado intersección, se realizó para tener un léxico conformado por palabras con una alta probabilidad de estar asociadas a un sentimiento, pues al estar presentes en todos los léxicos, nuestra hipótesis es que esas son las palabras polarizadas.

El segundo ensamble, llamado unión, se construyó bajo la idea de que al contener todas las palabras a las que se les ha asociado una carga positiva o negativa, puede apoyar a la clasificación; i.e., se busca tener mayor cobertura de vocabulario. La tabla 4 muestra los resultados en medida F de la clasificación usando los dos ensambles descritos anteriormente. En ambos casos, el valor

**Tabla 2.** Resultados en medida F de dos algoritmos de clasificación con tres diferentes esquemas de pesado para la colección en Inglés. Se usaron tres diferentes recursos léxicos: SentiWordNet, SenticNet y SentiCon.

SentiWordNet					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive Bayes	BOOL	0.710	<b>0.786</b>	0.778	0.780
	TF	0.680	0.696	0.711	0.710
	TF-IDF	0.683	0.696	0.711	0.710
SMO	BOOL	<b>0.735</b>	0.783	<b>0.782</b>	<b>0.802</b>
	TF	0.728	0.749	0.746	0.770
	TF-IDF	0.728	0.750	0.746	0.770
SenticNet					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive Bayes	BOOL	0.736	0.756	<b>0.770</b>	0.776
	TF	<b>0.766</b>	0.681	0.71	0.691
	TF-IDF	0.685	0.682	0.711	0.691
SMO	BOOL	0.744	<b>0.786</b>	0.769	<b>0.784</b>
	TF	0.722	0.727	0.739	76.7
	TF-IDF	0.722	0.728	0.74	0.767
SentiCon					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive Bayes	BOOL	0.699	<b>0.763</b>	0.735	<b>0.780</b>
	TF	0.676	0.656	0.667	0.667
	TF-IDF	0.676	0.659	0.665	0.668
SMO	BOOL	<b>0.737</b>	0.739	<b>0.755</b>	0.757
	TF	0.685	0.705	0.735	0.726
	TF-IDF	0.685	0.705	0.73.5	0.726

afectivo fue modificado al promedio de la suma de los valores en los léxicos involucrados. Nótese que este valor de sentimiento asociado a las palabras en los ensambles no se utiliza en este experimento.

De acuerdo con los resultados mostrados en la tabla 4, los mejores resultados fueron ligeramente menores a los que se obtuvieron en el primer experimento, usando únicamente el recurso léxico SentiWordNet. Se puede observar que el mejor desempeño fue de 0.79 de medida F y se obtuvo bajo el dominio Artículos de hogar con el algoritmo SMO en ambos casos.

Para el idioma español, se hicieron ambos ensambles: intersección y unión, con los dos recursos disponibles para este idioma. La tabla 5 muestra los resultados obtenidos para la colección en español. Se puede observar que con el ensamble unión se llegaron a resultados similares, aunque ligeramente mejores, que el mejor resultado del primer experimento. Se observa que, en el mejor caso, aquí se obtuvo 0.712 contra 0.707 del primer experimento, ambos bajo el algoritmo Naive Bayes con un esquema de pesado booleano. Por otro lado, con



**Tabla 3.** Resultados en medida F de dos algoritmos de clasificación con tres diferentes esquemas de pesado para la colección en Español. Se usaron tres diferentes recursos léxicos: ANEW y ML-SentiCon

		Anew	ML-SentiCon
Naive Bayes	BOOL	0.628	0.685
	TF	0.682	0.656
	TF-IDF	0.580	0.658
SMO	BOOL	0.643	<b>0.707</b>
	TF	0.632	0.694
	TF-IDF	0.633	0.694

**Tabla 4.** Resultados en medida F de dos algoritmos de clasificación con tres diferentes esquemas de pesado para la colección en Inglés. Se usaron dos ensambles de recursos léxicos: intersección y unión.

Intersección					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive Bayes	BOOL	0.709	<b>0.764</b>	0.719	0.750
	TF	0.670	0.643	0.645	0.644
	TF IDF	0.670	0.643	0.644	0.644
SMO	BOOL	<b>0.733</b>	0.748	<b>0.749</b>	<b>0.775</b>
	TF	0.688	0.710	0.737	0.739
	TF IDF	0.681	0.710	0.737	0.739
Unión					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive Bayes	BOOL	0.727	<b>0.792</b>	0.781	<b>0.791</b>
	TF	0.690	0.711	0.722	0.717
	TF IDF	0.689	0.711	0.721	0.717
SMO	BOOL	<b>0.756</b>	0.774	<b>0.787</b>	<b>0.791</b>
	TF	0.741	0.757	0.759	0.772
	TF IDF	0.741	0.757	0.759	0.772

el ensamble procedente de la intersección se llegó a, básicamente, los mismos resultados que en el experimento uno.

### 5.5. Evaluando la importancia de la carga afectiva de los recursos léxicos

Hasta el momento sólo se han usado los recursos léxicos para obtener el vocabulario a utilizar en la representación de los documentos. La idea detrás de este tercer experimento es determinar si la incorporación del valor afectivo que está asociada a cada palabra de cada recurso léxico es útil para mejorar el desempeño de la clasificación.

Para usar este valor afectivo se hicieron dos propuestas, la primera, multiplicar el valor booleano por el valor afectivo; la segunda, multiplicar la frecuencia de

**Tabla 5.** Resultados en medida F de dos algoritmos de clasificación con tres diferentes esquemas de pesado para la colección en Español. Se usaron dos ensambles de recursos léxicos: intersección y unión.

		Intersección Unión	
Naive Bayes	BOOL	0.685	0.694
	TF	0.656	0.673
	TF-IDF	0.658	0.672
SMO	BOOL	<b>0.707</b>	<b>0.712</b>
	TF	0.694	0.697
	TF-IDF	0.694	0.697

término por el valor en el léxico. En este experimento sólo se usaron los léxicos que obtuvieron los mejores resultados en los experimentos previos, i.e., SentiWordNet para inglés y el ensamble intersección para el español. Los resultados del desempeño del método de clasificación se muestran en la tabla 6.

**Tabla 6.** Resultados en medida F de dos algoritmos de clasificación con dos diferentes esquemas de pesado para la colección en Inglés. Se usó el recurso léxico SentiWordNet.

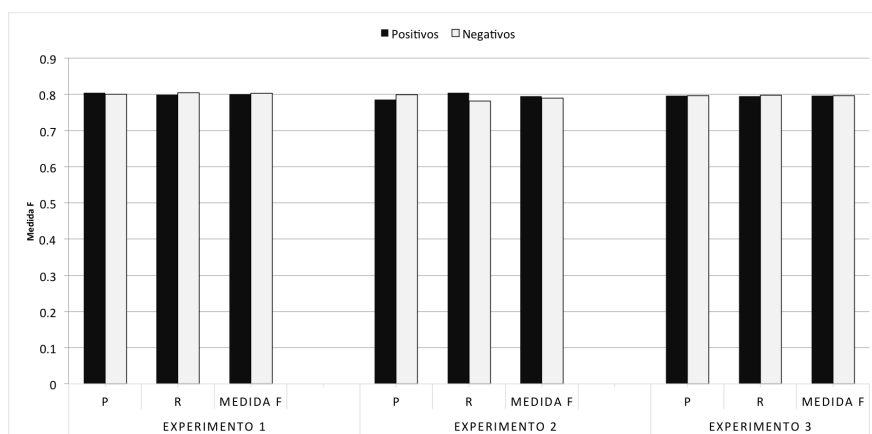
Sentiwordnet (Valor Afectivo)					
		Libros	DVD	Electrónicos	Artículos de hogar
Naive	BOOL	0.717	<b>0.783</b>	<b>0.780</b>	0.779
Bayes	TF	0.679	0.703	0.726	0.720
SMO	BOOL	0.738	0.770	0.769	<b>0.796</b>
	TF	<b>0.73</b>	0.748	0.749	0.777

Para la colección en inglés el mayor medida F es 0.796 contra 0.800 de la evaluación de SentiWordNet (ver Tabla 2), también bajo el dominio artículos de hogar con el algoritmo SMO y con el esquema de pesado que resultó de multiplicar el valor afectivo por el valor booleano. Para el caso de la colección en español se obtuvieron básicamente los mismos resultados que en experimentos previos (ver Tabla 7).

**Tabla 7.** Resultados en medida F de dos algoritmos de clasificación con dos diferentes esquemas de pesado para la colección en español. Se usó el recurso léxico ensamble intersección.

Ensamble (Valor Afectivo)		
Cine		
Naive	BOOL	0.685
Bayes	TF	0.664
SMO	BOOL	<b>0.707</b>
	TF	0.695

En general, los resultados producidos en los experimentos realizados se pueden considerar convenientes para el análisis de sentimientos. Entre los algoritmos utilizados se pueden apreciar mejores resultados de SMO sobre Naive Bayes, y por otro lado, de los esquemas de pesado propuestos, destaca booleano, incluido cuando se involucra el valor afectivo en los léxicos. Aunque, en la mayoría de los experimentos se llegaron a resultados muy parecidos. Por lo tanto, de estos experimentos se puede concluir que no importa ni el valor afectivo de las palabras ni el pesado diferente al booleano, es decir, es suficiente con indicar que una palabra del recurso léxico está o no presente en el documento. En la figura 2 y 3 se muestra un resumen de los mejores resultados obtenidos por los tres experimentos, indicando además valores de medida F por cada clase. Como puede observarse, los resultados de clasificación para ambas clases son similares por lo tanto las diferencias entre precisión, recuerdo y medida F son consistentes entre colecciones de datos.

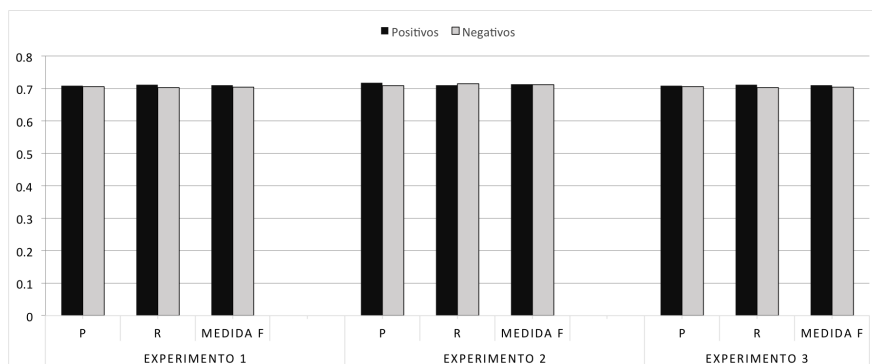


**Fig. 2.** Resumen de los mejores resultados de clasificación obtenidos para la colección en Inglés.

Otra conclusión importante es que el recurso léxico SentiWordNet es el más adecuado para diferentes dominios de revisiones de productos. Por otro lado, el ensamble intersección para español resultó ser mejor la clasificación de revisiones de películas. Es necesario realizar más experimentos con otras colecciones de datos para poder llegar a conclusiones más generales.

## 6. Conclusiones y trabajo a futuro

En este artículo se desarrolló un método de clasificación supervisado usando recursos léxicos para la tarea del análisis de sentimientos. La idea general de nuestro método es utilizar como representación el vocabulario de los documentos etiquetados que aparecen en ciertos recursos léxicos.



**Fig. 3.** Resumen de los mejores resultados de clasificación obtenidos para la colección en Español.

Además de evaluar diferentes recursos léxicos en diferentes idiomas, se trataron dos nuevas propuestas de recursos léxicos. La primera en la que se plantea dos ensambles que resultaron de la unión y la intersección de los léxicos disponibles. La segunda propuesta es utilizar el valor afectivo de los recursos léxicos como parte del pesado en la representación vectorial.

Con la primera propuesta de los ensambles de recursos léxicos se tenía la idea de que al tener mayor cobertura en los documentos se podrían obtener resultados más precisos, sin embargo, no resultó ser así en todos los casos, incluso se manifestaron resultados inferiores. De manera general, los resultados conseguidos no mostraron una diferencia significativa a los obtenidos de los clasificadores construidos con los léxicos. Se observa que se pueden llegar al mismo o, ligeramente, mejor rendimiento si se usan sólo los léxicos con un vocabulario de menor extensión.

Por otro lado, en la segunda propuesta se intentó comprobar cuanto afecta el valor afectivo de las palabras. Se llegó a resultados muy parecidos a los mostrados en el resto de experimentos. A partir de esto, se puede concluir que la información en los léxicos no afectan de forma importante.

Finalmente, se percata que las diferentes propuestas tienen un grado de fiabilidad bastante similar, también puede verse que en todos los experimentos se llegó a tener un rendimiento efectivo.

Como trabajo futuro se pretende primero hacer una evaluación de significancia estadística con los resultados obtenidos en este artículo. Además se piensan replicar el experimentos con otros recursos léxicos y otras colecciones de datos.

**Agradecimientos.** El trabajo de los dos últimos autores fue parcialmente financiado por el proyecto CONACyT CB-2015 No. 258588. También agradecemos al Departamento de Tecnologías de la Información de la Universidad Autónoma Metropolitana Unidad Cuajimalpa por el apoyo otorgado para la realización de este trabajo.

## Referencias

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
2. Blitzer, J., Dredze, M., Pereira, F., et al.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL. vol. 7, pp. 440–447 (2007)
3. Bradley, M.M., Lang, P.J.: Affective norms for english words (anew): Instruction manual and affective ratings. Tech. rep. (1999)
4. Cruz, F., Troyano, J., Enriquez, F., Ortega, J.: Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento del lenguaje Natural* 41, 73–80 (2008)
5. Cruz Mata, F., Troyano Jiménez, J.A., Pontes Balanza, B., Ortega Rodríguez, F.J.: Ml-senticon: un lexicón multilingüe de polaridades semánticas a nivel de lemas (2014-09)
6. Devaraj, M., Piryani, R., Singh, V.K.: Lexicon ensemble and lexicon pooling for sentiment polarity detection. *IETE Technical Review* 33(3), 332–340 (2016), <https://doi.org/10.1080/02564602.2015.1073572>
7. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: *Proceedings of the 2008 international conference on web search and data mining*. pp. 231–240. ACM (2008)
8. Garner, S.R.: Weka: The waikato environment for knowledge analysis. In: *In Proc. of the New Zealand Computer Science Research Students Conference*. pp. 57–64 (1995)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. pp. 79–86. Association for Computational Linguistics (2002)
10. Real-Flores, G., García-Mendoza, B., Calderón-Casanova, E., de-la Rosa, G.R., Villatoro-Tello, E.: Generación y enriquecimiento automático de recursos léxicos para el análisis de sentimientos. In: *Research in Computing Science*. vol. 134, pp. 163–176 (2017)



## Arquitectura para el análisis de estados financieros XBRL publicados por empresas en México utilizando lógica difusa

Cristian Noé Enríquez-Marcial<sup>1</sup>, Hilarion Miño-Contreras<sup>1</sup>,  
José Luis Sánchez-Cervantes<sup>2</sup>, Lisbeth Rodríguez-Mazahua<sup>1</sup>, Giner Alor-Hernández<sup>1</sup>

<sup>1</sup> División de estudios de Posgrado en Investigación,  
Instituto Tecnológico de Orizaba,  
México

<sup>2</sup> CONACYT – Instituto Tecnológico de Orizaba,  
México

{marcial2005, lisbeth08}@gmail.com, hmunoz189@msn.com, ginalor@outlook.com,  
jsanchez@ito-depi.edu.mx

**Resumen.** El análisis de estados financieros es el proceso dirigido a evaluar la posición financiera, así como los resultados de las operaciones de una empresa. Los más usados son: el balance general y el estado de resultados. El surgimiento de la inteligencia artificial dio origen a sistemas que tienen la capacidad de realizar tareas complejas y con la capacidad de auxiliar a un usuario en un área específica, dando respuestas tal y como un experto lo haría. Las condiciones presentes en los mercados financieros dieron origen al desarrollo de herramientas que proporcionen información sobre la salud financiera de una empresa, necesitando modelos innovadores que logren dar solución a problemas que no pueden resolverse con métodos clásicos, sino con aquellos que proporciona la inteligencia artificial y la lógica difusa, ya que proporcionan los medios para apoyar la toma de decisiones de manera oportuna y precisa. En este artículo se propone el diseño de la arquitectura de un sistema experto basado en lógica difusa para el análisis de estados financieros XBRL de empresas mexicanas como apoyo en la toma de decisiones. La arquitectura se basa en un diseño de capas donde cada funcionalidad de los componentes se distribuye homogéneamente para facilitar la escalabilidad de la aplicación web.

**Palabras clave:** inteligencia artificial, lógica difusa, XBRL, análisis financiero, estados financieros, sistema experto, Web.

### Architecture for the Analysis of XBRL Financial Statements Published by Companies in México using Fuzzy Logic

**Abstract.** The analysis of financial statements is the process aimed at evaluating the financial position, as well the results of the operations of a company, the most used are: The balance sheet and the income statement. The emergence of artificial

intelligence gave rise to systems that have the ability to perform complex tasks and with the ability to assist a user in a specific area, giving answers as an expert would. The present conditions in the financial markets gave rise to the development of tools that provide information on the financial health of a company, needing innovative models that manage to solve problems that can not be solved with classical methods, if not, with those that provide intelligence, artificial and fuzzy logic, as they provide the means to support decision making in a timely and accurate manner. This article proposes the design of the architecture of an expert system based on fuzzy logic for the analysis of XBRL financial statements of Mexican companies as support in decision making. The architecture is based on a layer design where each functionality of the components is distributed homogeneously to facilitate the scalability of the web application.

**Keywords:** artificial intelligence, fuzzy logic, XBRL, financial analysis, financial statements, expert system, Web.

## **1. Introducción**

El análisis de estados financieros es el proceso dirigido a evaluar la posición financiera, así como los resultados de las operaciones de una empresa, con el objetivo de establecer las mejores estimaciones y predicciones posibles sobre las condiciones y resultados futuros. El proceso de análisis de estados financieros consiste en la aplicación de herramientas y técnicas analíticas a los estados y datos financieros, con el fin de obtener de ellos medidas y relaciones que son significativas y útiles para la toma de decisiones.

Así, el análisis financiero cumple en primer lugar la función de convertir los datos en información útil [1]. Adicionalmente, durante el análisis de los estados financieros se identifican aspectos relevantes para el apoyo en la toma de decisiones. Desarrollar un análisis ayuda a evaluar el valor de los estados financieros de una empresa [2], complementando esta aseveración en [3] mencionan que el análisis fundamental tiene por objeto determinar el valor de los títulos privados mediante un examen cuidadoso de factores clave/valor, tales como los ingresos, las inversiones, el riesgo, el crecimiento y la posición competitiva, entre otros.

La lógica difusa está relacionada y fundamentada en la teoría de conjuntos difusos, según la cual el grado de pertenencia de un elemento a un conjunto está determinado por una función de pertenencia que puede tomar todos los valores reales comprendidos en el intervalo de (0, 1) [4].

Cuando la información es imprecisa o insuficiente, usar instrumentos estadísticos no basta para obtener resultados significativos. La lógica difusa surge precisamente para tratar con este tipo de problemas y lograr darles una solución óptima. De esta forma, una combinación entre un sistema de lógica difusa y la experiencia o conocimiento que tienen los encargados de tomar las decisiones es una excelente manera de obtener buenos resultados.

La aplicación de la lógica difusa permite enfrentar problemas de manera efectiva para la creación de sistemas de soporte para la toma de decisiones, ya que los modelos que se utilizan son altamente flexibles y más tolerantes a la imprecisión de los datos. Por otra parte, una razón financiera es un indicador esencial para conocer la situación



económica de una empresa [5], y se expresa por medio de una fórmula matemática específica y simple.

Las razones financieras proporcionan información que beneficia a los interesados en la toma de decisiones empresariales. Además, ayudan a determinar la magnitud y la dirección de los cambios en la empresa durante un período de tiempo. Actualmente, la estandarización se ha convertido en un problema importante para el desarrollo de varias industrias. Los estándares propician mejorar la calidad de los datos de la industria. En este sentido, XBRL (eXtensible Business Reporting Language), un lenguaje basado en XML (eXtensible Markup Language) que permite la publicación electrónica de datos comerciales y financieros, es aplicado a muchos dominios, incluida la supervisión financiera, regulación gubernamental y control interno de la corporación [6].

Es imperativo mencionar que en México a partir del primer trimestre de 2016 las emisoras industriales, comerciales y de servicios tienen la obligación de enviar su información financiera trimestral en formato XBRL [6]. Para finales de 2017 fueron publicadas nuevas taxonomías, por lo que se estimó que la Comisión Nacional Bancaria y de Valores (CNBV) publicara como fecha de inicio de la obligación para la publicación de reportes bajo este nuevo esquema a partir del 1 de enero de 2018 [7]. Como se puede observar, la adopción e implementación del estándar XBRL en México está en proceso de consolidación.

Considerando lo anterior, la principal contribución de este trabajo es proporcionar una arquitectura para un sistema experto basado en lógica difusa que realice el análisis de los estados financieros publicados en formato XBRL por empresas mexicanas, con el propósito de generar un sistema de análisis financiero que brinde soporte a la información publicada en formato XBRL, apoyado de la lógica difusa para la interpretación de resultados y como apoyo a la toma de decisiones financieras.

Este documento está estructurado de la siguiente manera: La Sección 2 presenta un conjunto de trabajos relacionados con esta propuesta, organizándolos en los temas que presentan el análisis financiero usando lógica difusa y el de análisis financiero con XBRL; en la Sección 3, se describe la arquitectura propuesta, las capas que la integran, los componentes de cada capa, así como el flujo de trabajo entre ellos; además, la sección 4 incluye un estudio de caso y, finalmente, la sección 5 presenta las conclusiones y el trabajo futuro.

## **2. Trabajos relacionados**

A continuación, se presenta una revisión del estado del arte en la que se toman en cuenta los aspectos del análisis financiero, la lógica difusa y el estándar XBRL.

### **2.1. Análisis financiero y lógica difusa**

En [8] se desarrolló un sistema experto que soluciona problemas de autorización de crédito bancario. El sistema utiliza los resultados del análisis de crédito de los usuarios para crear una base de conocimientos cuyas entradas y parámetros se caracterizan por valores difusos, por esta razón el uso de un mecanismo de inferencia de lógica difusa se utilizó en el sistema para apoyar en la toma de decisiones para otorgar créditos bancarios.

En [9] se proporcionó un sistema experto estadístico difuso para la gestión del flujo de efectivo. El sistema ayudó a administrar los recursos efectivos de la organización. Para ello, los autores definieron las variables de entrada, salida y sus funciones de pertenencia, se formaron reglas usando el sistema de inferencia difuso para inferir el saldo de efectivo final de un conjunto de combinación de 25 reglas separadas. Finalmente, los niveles lingüísticos se convirtieron a ciertos números por el método centrado para ayudar a ver los efectos de los cambios en los niveles de insumos en los saldos de efectivo finales.

En [10] se estudió y modeló un sistema de asignación de recursos financieros a compañías comisionistas de bolsa, con el fin de disminuir el riesgo de impago del capital asignado y, además; que estos capitales generen rendimientos adicionales para la empresa. El modelo planteado basado en sistemas expertos difusos permitió soportar estas decisiones de asignación de recursos financieros.

Arias-Aranda, et al. [11], diseñaron un sistema experto difuso centrado en aumentar la precisión y la calidad del conocimiento para la toma de decisiones. El sistema utilizó un modelo basado en reglas difusas para simular el comportamiento de las empresas, se presentó bajo el supuesto de parámetros de entrada determinados previamente detectados y se desarrolló un algoritmo para lograr la estructura mínima del modelo. El resultado obtenido fue una herramienta de sistema experto difuso, llamada ESROM, que proporciona información valiosa para ayudar a los gerentes a mejorar el logro de los objetivos de la empresa. Una de las principales aportaciones de la iniciativa presentada por los autores de este trabajo es que el sistema es general, y se adapta a diferentes escenarios.

En [12] se propuso la adopción de una técnica inteligente híbrida, un sistema experto difuso para llevar a cabo un análisis costo-beneficio de la inversión en sistemas de información empresarial. Los modelos tradicionales de presupuestación de capital se centran en variables cuantificables. Sin embargo, hay muchas variables intangibles que hacen que el uso de medidas completamente cuantitativas sea incompleto y menos inclusivo. Por lo que en este estudio se tomó gran conocimiento de las variables intangibles y de vaguedad en la toma de decisiones del grupo humano que requiere un alto nivel de consenso.

En los sistemas expertos existentes la incertidumbre se trata mediante una combinación de lógica predicativa y métodos basados en la probabilidad. Una deficiencia grave de estos métodos es que no son capaces de enfrentarse a la información difusa en la base de conocimientos. Por lo que un enfoque alternativo al manejo de la incertidumbre se sugirió en [13], el cual se basa en el uso de la lógica difusa.

En [14] se planteó un enfoque híbrido para la evaluación del desempeño financiero de las compañías automotrices de la bolsa de Teherán. Para ello, se estructuró un modelo jerárquico de evaluación del desempeño financiero basado en las medidas contables y las medidas de valor económico. En este enfoque se aplicó el Fuzzy Analytic Hierarchy Process para determinar los pesos de los criterios, también los resultados de tres métodos de superación se combinaron usando los rangos medios. Los resultados representaron que las medidas de valor económico son más importantes que las medidas contables en la evaluación del desempeño financiero de las empresas.

Los casos asociados con el análisis de riesgo del préstamo y las estrategias de adaptación relacionadas han crecido en importancia y complejidad, por lo que Kumar,

Bhatia y Kapoor [15] plantearon controlar este problema utilizando el conocimiento y la experiencia de expertos involucrados en este proceso y que tienen conocimiento sobre este campo. Por tal razón dichos autores proporcionaron el análisis de riesgo basado en un sistema de inferencia difusa, así como la interfaz gráfica de usuario que consideró los diferentes parámetros de un solicitante. El análisis de los resultados experimentales de los diferentes solicitantes comprobó la corrección y la coherencia de la decisión del sistema de apoyo para la toma de decisiones.

## **2.2. Análisis financiero con XBRL**

El número de aplicaciones de software que permiten analizar informes XBRL es escaso, debido a la reciente estandarización de los informes financieros que utilizan XBRL, lo que la convierte en un área de estudio interesante y una fuente de oportunidades de investigación aprovechada por S. Mendez et al. [16], quienes agruparon estas aplicaciones en las dos categorías siguientes: 1) aplicaciones para la validación, edición y generación de informes contables, casi todas las aplicaciones de este tipo son privadas y tienen altos costos de licencia y, 2) un conjunto de herramientas que ayudan a los usuarios de los informes XBRL a evaluar la solvencia y la rentabilidad futura de las empresas en estudio. En [16] se propuso un enfoque para extraer información de conjuntos de documentos que cumplen con diferentes taxonomías XBRL en diferentes formatos. Esto se hace clasificando la información contable en una ontología de conceptos financieros diseñados para este propósito. El procesamiento automatizado de la información, que es posible debido al uso de XML, brinda a los usuarios la máxima flexibilidad en sus consultas.

En [17] se presentó una plataforma inteligente llamada FLORA, la cual provee un enfoque para hacer frente a las deficiencias actuales de información financiera y lograr una forma más efectiva de procesar datos financieros basados en los principios de Linked Data.

En éste se describió el proceso de extracción de datos y el modelado semántico que son los pilares del análisis de datos financieros. Como resultado, FLORA facilita un análisis financiero eficaz, basado en datos, y una integración a escala web entre las aplicaciones financieras y las plataformas.

## **2.3. Análisis comparativo de los trabajos relacionados**

A continuación, se presenta un breve análisis comparativo entre los trabajos relacionados y nuestra iniciativa. La Tabla 1 indica si el trabajo presentado describe su arquitectura, el tipo de estados financieros que analiza, el tipo de análisis que se lleva a cabo sobre la información de origen, y si ofrecen soporte al estándar XBRL.

Como se observa en la Tabla 1, el desarrollo e investigación de sistemas expertos para el análisis financiero, utilizando lógica difusa, es cada vez más extendido y ocurre en diferentes contextos como en [8, 9, 10 y 15] que se enfocan en el análisis de riesgos para la autorización de créditos y préstamos, la asignación de recursos; sin embargo, es importante destacar que pocas iniciativas de las analizadas en la literatura presentan una arquitectura que permita la integración de lógica difusa como parte esencial del análisis de estados financieros publicados bajo el estándar XBRL como en [16] y [17];

**Tabla 1.** Análisis comparativo entre los trabajos relacionados y el presente trabajo.

Iniciativa	Arquitectura	Contexto del análisis financiero	Tipo de análisis	Soporte XBRL
M. Menekay. [8]	No	Autorización de créditos bancarios	Sistema de inferencia difuso	No
A. Anvary Rostamy et. al. [9]	No	Flujo de efectivo	Sistema de inferencia difuso	No
S. Medina Hurtado et. al. [10]	No	Asignación de recursos financieros a compañías comisionistas	Sistema de inferencia difuso	No
S. Kumar et. al. [15]	No	Análisis de riesgo del préstamo	Sistema de inferencia difuso	No
S. Mendez Nuñez et. al. [16]	Sí	Balance general Estado de resultados	Tecnologías de Web Semántica	Sí
M. Radzimski et. al. [17]	Sí	Balance general Estado de resultados Flujo de efectivo	Tecnologías de Web Semántica	Sí
<b>Nuestra iniciativa</b>	Sí	Balance general Estado de resultados Flujo de efectivo	Sistema de inferencia difuso	Sí

no obstante, estas iniciativas no analizan estados financieros publicados en México y no incluyen lógica difusa para la generación de recomendaciones financieras.

Adicionalmente, nuestra iniciativa incluye en el diseño una arquitectura organizada en capas para su fácil mantenimiento, misma que se es descrita brevemente en la siguiente sección.

### 3. Arquitectura

La arquitectura que se muestra en la Fig. 1, se basa en cuatro capas. Cada capa contiene componentes, algunos de ellos se conforman por subcomponentes. Las tareas y responsabilidades de la aplicación se distribuyen entre sus distintos componentes. La arquitectura diseñada permite fácil escalabilidad y mantenimiento.

Asimismo, la arquitectura mostrada en la Fig. 1, ayuda a comprender gráficamente los componentes estructurales de los cuales se integra las relaciones entre ellos, así como el flujo de trabajo que se llevará a cabo para la comunicación y la obtención de resultados entre componentes.

#### 3.1. Descripción de capas

Cada una de las capas tiene una función explicada a continuación:

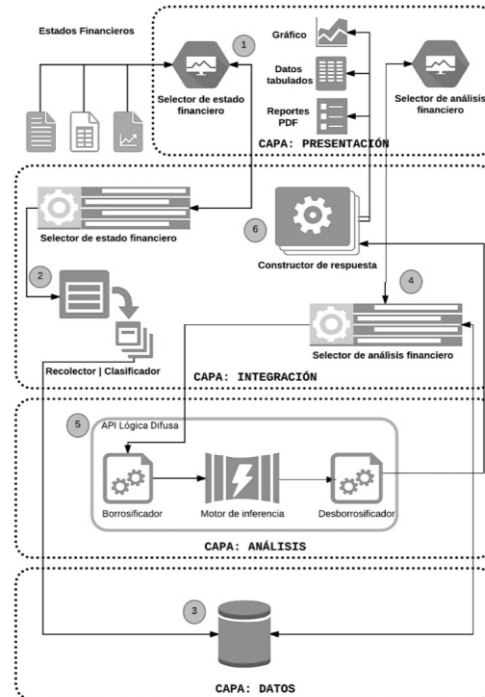


Fig. 1. Arquitectura del sistema experto difuso.

- **Capa de presentación:** Esta capa representa la interfaz entre el usuario y la aplicación. Contiene los componentes que hacen posible la interacción del usuario con la aplicación.
- **Capa de integración:** Esta capa contiene los componentes necesarios para enviar y recibir los datos e interactuar con la capa de presentación. Esta capa incluye el selector de estado financiero, así como la construcción de la información que será mostrada al usuario.
- **Capa de análisis:** Esta capa contiene la Interfaz de Programación de Aplicaciones (API) de lógica difusa, que será la encargada de borrosificar los datos para posteriormente realizar el análisis financiero y entregará un resultado, el cual será entregado al constructor de respuesta.
- **Capa de datos:** Para esta arquitectura ha sido considerado tener un repositorio de información para almacenar los datos que han sido extraídos y clasificados para su posterior análisis, así como almacenar los resultados de los análisis realizados.

### 3.2. Descripción de componentes

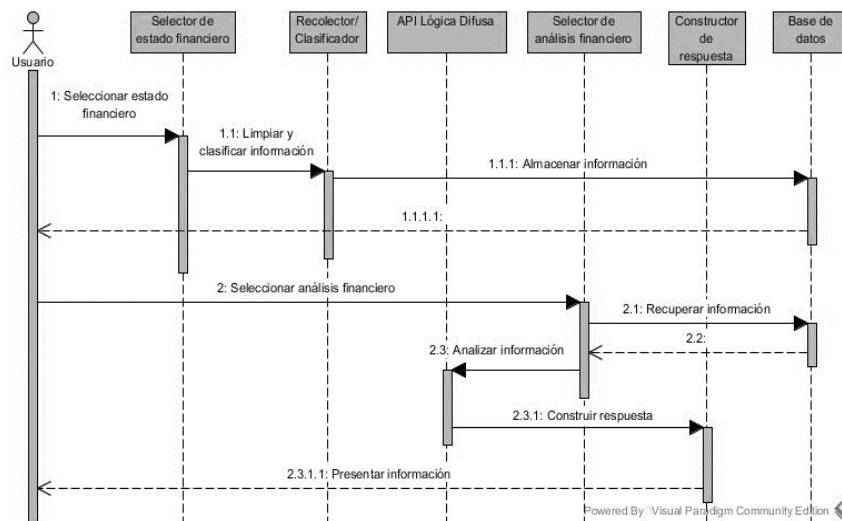
Los componentes que forman parte de cada capa de la arquitectura (Fig. 1), tienen funciones que definen su comportamiento, el cual se describe brevemente a continuación:

- **Selector de estado financiero:** Proporciona una Interfaz Gráfica de Usuario (GUI) desarrollada a través del marco de trabajo Spring MVC (Modelo Vista Controlador) y componentes de PrimeFaces, este último para el diseño responsivo de la página para que se adapte a la pantalla del dispositivo que la visualiza. A través de este componente, el usuario tiene la opción de elegir el estado financiero que será objeto de análisis por el sistema, la GUI proporciona la interfaz para que el usuario provea la información de los estados financieros en formato XBRL, Valores separados por Comas (CSV), y la Notación de objetos de JavaScript (JSON), entre otros.
- **Recolector:** Es el encargado de recuperar y limpiar la información enviada por el usuario mediante el componente selector de estado financiero.
- **Clasificador:** Una vez hecha la limpieza de la información por el módulo de recolección, este componente se encargará de realizar la clasificación de los datos para almacenarlos en el componente base de datos.
- **Selector de análisis financiero:** A través de este componente, el usuario tiene la opción de elegir el tipo de análisis financiero que será realizado por el sistema a la información obtenida de la base de datos, la cual será proporcionada al componente de la API de lógica difusa.
- **Constructor de respuesta:** Este componente recibe las respuestas de la API de lógica difusa después de la ejecución del análisis seleccionado, recupera la información y crea un reporte de análisis financiero, dicho documento se presentará en formato gráfico visual, en un reporte PDF o en datos tabulados para apoyar al usuario en la toma de decisiones.
- **API Lógica difusa:** Este componente recibe la información recuperada por el selector de análisis financiero, para realizar su procesamiento. Su función principal es realizar la borrosificación de los datos recuperados para después, realizar el análisis mediante el motor de inferencia donde previamente se han definido las reglas y conjuntos de pertenencia de la información, una vez analizada dicha información se procede a la desborrosificar los datos para obtener las razones financieras de análisis y generar una respuesta mediante el constructor de respuesta.
- **Base de datos:** La función de este componente es almacenar la información que mediante un proceso de extracción, limpieza y transformación es recuperada por el módulo recolector/clasificador de la aplicación. De la misma manera que almacena los resultados generados por el selector de análisis financiero.

### **3.3. Flujo de trabajo**

Las relaciones entre los componentes de la arquitectura que se muestran en la Fig. 1 definen el flujo de trabajo para el proceso del análisis de estados financieros, desde la selección del tipo de dato de entrada, hasta la elección de como el usuario desea se presenten los resultados. Aquí hay una breve descripción del flujo de trabajo de la arquitectura:

1. A través de la GUI desarrollada usando el marco de trabajo Spring MVC, el usuario realiza una solicitud basada en el protocolo HTTP al ingresar mediante un selector de archivo situado en la capa de presentación, el estado financiero a analizar. El componente selector de estado financiero situado en la capa de integración envía el documento proporcionado por el usuario a componente recolector/clasificador.



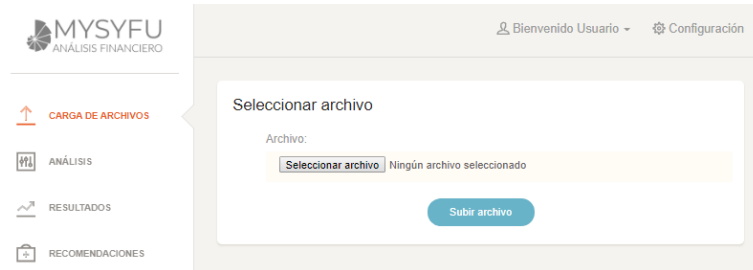
**Fig. 2.** Diagrama de secuencia del flujo de trabajo de la arquitectura.

2. La información será extraída del documento por el componente recolector/clasificador para realizar la limpieza y clasificación de los datos.
3. Los datos extraídos por el componente recolector/clasificador será almacenada en la base de datos.
4. Por medio del componente selector de análisis financiero situado en la capa de presentación, el usuario seleccionará el tipo de análisis que se aplicará a la información almacenada en la base de datos. El componente selector de análisis financiero situado en la capa de integración enviará el tipo de análisis a realizar y la información a la API de lógica difusa.
5. La API de lógica difusa hará la borrosificación de la información proporcionada, mediante el motor de inferencia, el cual contiene las reglas proporcionadas por un experto, será el encargado de realizar el análisis financiero y definir el conjunto de pertenencia de la información, la cual será desborrosificado para su interpretación.
6. La respuesta proporcionada por la API de lógica difusa será enviada al componente constructor de respuesta situado en la capa de integración que será el encargado de presentar la información de forma visual para el usuario y así apoyar en la toma de decisiones.

En la Fig. 2, se muestra un diagrama de secuencia basado en el Lenguaje Unificado de Modelado (UML) con una descripción detallada el flujo de trabajo descrito previamente.

#### 4. Estudio de caso

Para este caso se parte del supuesto de que una persona desea invertir capital en una empresa, por lo que es necesario realizar un análisis financiero. El análisis financiero



**Fig. 3.** Interfaz de carga de documentos XBRL.



MYSYFU

ANÁLISIS FINANCIERO

CARGA DE ARCHIVOS

ANÁLISIS

RESULTADOS

RECOMENDACIONES

Listado de empresas en el sistema

EMPRESA, S. A. B. DE C. V

Información financiera

Año	Trimestre	Activo Circulante	Pasivo Circulante	Inventario	Pasivo Total	Activo Total
2015						
	1	22045024	18892192	15806298	30977382	78158201
	2	22684324	18242583	16756134	30607762	78559662
	3	23242395	17802287	17004014	30040026	78864026
	4	46085122	22556102	17588048	51716061	10184476

**Fig. 4.** Información financiera extraída de los estados financieros XBRL.

**Tabla 1.** Fórmulas para calcular algunas razones financieras.

Ratio Financiero	Fórmula
<b>Razón corriente</b>	$\text{Razón corriente} = \frac{\text{Activo corriente}}{\text{Pasivo Circulante}}$
<b>Capital de trabajo</b>	$\text{Capital de trabajo} = \frac{(\text{Activo circulante} - \text{Pasivo Circulante})}{\text{Pasivo Circulante}}$
<b>Prueba acida</b>	$\text{Prueba Ácida} = \frac{(\text{Activo circulante} - \text{Inventarios})}{\text{Pasivos Circulantes}}$
<b>Razón de deuda</b>	$\text{Razón de deuda} = \left( \frac{\text{Pasivos totales}}{\text{Activos totales}} \right) * 100$

busca obtener una recomendación para invertir a partir del cálculo automático de los siguientes indicadores financieros:

- A) Razón corriente: Permite determinar el índice de liquidez de una empresa;
- B) Capital de trabajo: Busca garantizar las operaciones de la empresa, si el resultado es positivo, da la posibilidad de generar inversión y si es negativo, da la posibilidad de buscar financiamiento propio o mediante fondos de terceros;
- C) Prueba Ácida: Revela la capacidad de la empresa para cancelar sus obligaciones corrientes, pero sin contar con la venta de sus existencias y,



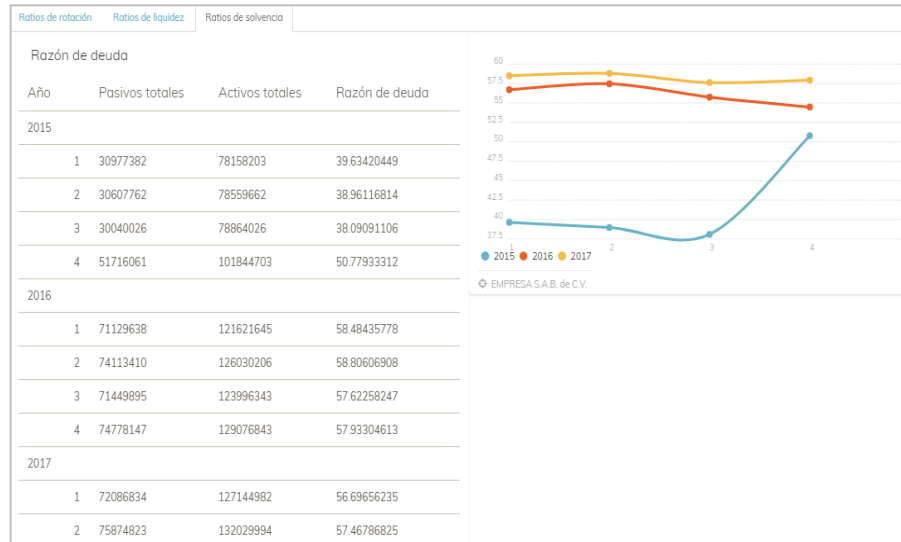


Fig. 5. Información del cálculo de las razones financieras.

Tabla 3. Indicador de Razón corriente por año.

Año	Activo Circulante	Pasivo Circulante	Razón Corriente	Obligaciones	Disponible
2015	114056775	77493164	1.471830147	67.94262244	32.05737756
2016	136758411	127231863	1.074875489	93.03403138	6.965968623
2017	151602198	146930010	1.031798732	96.91812648	3.081873523

Tabla 4. Indicador de Prueba ACID.

Año	Activo Circulante	Pasivo Circulante	Inventario	Prueba Ácida
2015	114056775	77493164	67155294	0.605233786
2016	136758411	127231863	96461883	0.316717268
2017	151602198	146930010	110481272	0.279867442

Tabla 5. Indicador de Razón de deuda.

Año	Pasivos totales	Activos totales	Razon de deuda
2015	143341231	337426594	42.48071538
2016	291471090	500725037	58.20980947
2017	290419217	517692020	56.09883981

D) Razón de deuda: permite establecer el grado de participación de los acreedores, en los activos de la empresa. Mide la proporción de la inversión de la empresa que ha

sido financiada por deuda, por lo que se acostumbra a presentar su resultado en forma de porcentaje.

Tales indicadores permiten agilizar el diagnóstico de la situación actual de la empresa y facilita la toma de su decisión para realizar una inversión. Con la solución propuesta, se pretende que la persona interesada pueda obtener una recomendación para realizar la toma de decisiones financieras, por lo que lleva a cabo el proceso siguiente:

A partir de un conjunto de datos proporcionados en formato XBRL se obtiene la información que permite realizar el análisis de los estados financieros de la empresa, como lo son el estado de situación financiera, el estado de resultados, el estado de flujo de efectivo, entre otros. En la Fig. 3, se muestra la interfaz para la carga de los estados financieros de la empresa a analizar.

De los datos extraídos se obtiene la información financiera del periodo comprendido entre los años 2015 y 2017 para la empresa, a partir de esta se calcularán las razones financieras aplicando las formulas que se muestra en la Tabla 2, una vez realizado el calculo de las razones, la información se visualiza como se muestra en la Fig. 4.

Las razones financieras son unos indicadores que profundizan en la información de los estados financieros, estos están catalogados en ratios de liquidez, solvencia, rendimiento y apalancamiento. En la Fig. 5, se presenta la información obtenida del cálculo de las razones.

A partir del valor obtenido para el cálculo de la razón corriente que se muestra en la Tabla 3 en el año 2015 se da la siguiente interpretación: por cada peso de obligación vigente (deuda), la empresa contaba con 1.47 pesos para respaldarla.

Con los valores obtenidos para la prueba acida, los cuales observamos en la Tabla 4, se obtuvo que en los años más recientes 2016 y 2017, la empresa mantuvo valores de entre 0.31 y 0.22 respectivamente, los cuales, están por debajo del valor óptimo indicado para esta prueba (0.50 a 1.0), lo que significa que la empresa, si en algún momento tuvo la necesidad de atender todas sus obligaciones corrientes sin la necesidad de liquidar y vender sus inventarios, a esta, no le habría alcanzado y, por lo tanto, habría tenido que vender sus inventarios (o parte de ellos) para poder cumplir con sus obligaciones.

Aplicando la formula dada anteriormente, se calcula la Razón de deuda de la empresa, de esta manera, analizan e interpretan los resultados obtenidos y mostrados por año en la Tabla 5., los cuales indican que para el año 2015, el 42% del total de la inversión (Activos Totales) fue financiado con la participación de sus acreedores.

Esta situación aumento para el 2016 por lo que representa un moderado nivel de riesgo inversión para la empresa.

El caso de estudio presentado muestra la realización del análisis de estados financieros a partir de su representación en formato XBRL, por lo que se continuará con el desarrollo de un módulo de lógica difusa que analizará los indicadores financieros de manera automática con el propósito de generar recomendaciones que permitan apoyar en la toma de decisiones al usuario final.

## **5. Conclusiones y trabajo futuro**

Los aspectos tratados en el presente trabajo servirán como base para el desarrollo de una herramienta de autoría. Se presentó una arquitectura que, mediante el uso de

tecnologías Web y lógica difusa facilitará el análisis de estados financieros publicados por las empresas mexicanas utilizando el estándar XBRL. Los datos contenidos en los informes XBRL son un gran avance para la estructuración y el análisis de la información financiera. La implementación de la arquitectura propuesta aumentaría la eficiencia de los procesos de toma de decisiones a través de la incorporación de un módulo de lógica difusa que permita generar recomendaciones con la finalidad de apoyar en la toma de decisiones financieras empresariales.

Como trabajo a futuro se implementarán dos módulos, el primero será el de lógica difusa que genere las recomendaciones automáticas necesarias para dar soporte a la toma de decisiones utilizando inferencia basada en reglas apoyado en la arquitectura propuesta, el segundo módulo permitirá exportar los resultados obtenidos en los formatos comunes incluyendo hojas de cálculo, PDF, JSON y XML.

Finalmente, se considera el uso de técnicas de aprendizaje profundo que permitan la generación de analítica predictiva para detectar interacciones en los datos financieros nuevos e históricos que pueden pasar desapercibidas, pero que ayudarán a obtener el conocimiento necesario para la creación de modelos predictivos que de manera automática, apoyen en la toma de decisiones para predecir el comportamiento y tendencias de las empresas con base en la información expuesta en sus estados financieros.

**Agradecimientos.** Los autores de este documento agradecen el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT) así como al Tecnológico Nacional de México (TecNM) por el apoyo para la realización de este trabajo.

## Referencias

1. Bernstein, L.A.: Análisis de estados financieros: teoría, aplicación e interpretación, 657.31/B53fE (1995)
2. Ou, J.A., Penman, S.H.: Financial statement analysis and the prediction of stock returns. *J. Account. Econ.*, 11(4), pp. 295–329 (1989)
3. Lev, B., Thiagarajan, S.R.: Fundamental information analysis. *J. Account. Res.*, pp. 190–215 (1993)
4. Kulkarni, A. D.: Computer vision and fuzzy-neural systems. Prentice Hall PTR (2001)
5. Bliss, J.H.: Financial and operating ratios in management. The Ronald press company (1923)
6. Santos-Morales, R.A.R.J.N., Muguera-Medina L.: Bolsa Mexicana de Valores (2016) [https://www.bmv.com.mx/work/models/Grupo\\_BMV/Resource/1928/Presentacion\\_XBRL\\_Portal\\_10feb16\\_7.56.pdf](https://www.bmv.com.mx/work/models/Grupo_BMV/Resource/1928/Presentacion_XBRL_Portal_10feb16_7.56.pdf) (2018)
7. 2H Software: XBRL México. <http://www.xbrl.mx/Noticias2HTaxonomia.aspx> (2018)
8. Menekay, M.: Bank Credit Authorization Using Fuzzy Expert System. *Procedia Comput. Sci.*, 102, pp. 659–662 (2016)
9. Rostamy, A.A., Baghaei,V.F., Takanlou, B., Rostamy, A.A.: A fuzzy statistical expert system for cash flow analysis and management under uncertainty. *Adv. Econ. Bus.*, 1(2), pp. 89–102 (2013)
10. Medina-Hurtado, S., Manco, O.O.: Design of a fuzzy expert system: Credit risk assessment of stock brokerage firms in granting financial resources. *Estud. gerenciales*, 23(104), pp. 101–129 (2007)
11. Arias-Aranda, D., Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: A fuzzy expert system for business management. *Expert Syst. Appl.*, 37(12), pp. 7570–7580 (2010)

12. UzokaF, E.: Fuzzy-expert system for cost benefit analysis of enterprise information systems: a framework. *Int. J. Comput. Sci. Eng.*, 1(3), pp. 254–262 (2009)
13. Zadeh, L.A.: The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy sets Syst.*, 11(1–3), pp. 199–227 (1983)
14. Safaei-Ghadikolaei, A., Khalili Esbouei, S., Antucheviciene, J.: Applying fuzzy MCDM for financial performance evaluation of Iranian companies. *Technol. Econ. Dev. Econ.*, 20(2), pp. 274–291 (2014)
15. Kumar, S., Bhatia, N., Kapoor, N.: Fuzzy logic based decision support system for loan risk assessment. In: *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*, pp. 179–182 (2011)
16. Núñez, S.M., Andrés-Suárez, J., Gayo, J.E.L., De Pablos, P.O.: A semantic based collaborative system for the interoperability of XBRL accounting information. In: *World Summit on Knowledge Society*, pp. 593–599 (2008)
17. Radzinski, M., Sanchez-Cervantes, J.L., Garcia-Crespo, A., Temiño-Aguirre, I.: Intelligent architecture for comparative analysis of public companies using semantics and XBRL data. *Int. J. Softw. Eng. Knowl. Eng.*, 24(5), pp. 801–823 (2014)

## **Desarrollo de un sistema domótico con controlador difuso y controlador manual, implementado en LabView y Arduino IDE**

José Alberto Vázquez Fernández, David Tinoco Varela

Universidad Nacional Autónoma de México,  
Facultad de Estudios Superiores Cuautitlán, Departamento de Ingeniería,  
Ingeniería en Telecomunicaciones, Sistemas y Electrónica,  
México

itse.avazquez@gmail.com, dativa19@hotmail.com

**Resumen.** En este trabajo se abordan los aspectos de diseño, implementación y experimentación de un sistema domótico, este esquema ha sido puesto en funcionamiento por medio de las herramientas de desarrollo LabView y Arduino IDE. El sistema busca gestionar aspectos comunes dentro de un hogar mediante controladores difusos y controladores manuales, con la intención de que el inmueble diseñado pueda tener una respuesta automática, pero que también pueda ser controlado a gusto del usuario ante situaciones que salgan de lo cotidiano. En este sistema se considera la seguridad del inmueble mediante la implementación de una interfaz que pueda monitorizar la edificación a través de una cámara I P. Este trabajo fue realizado, buscando generar ambientes confortables, y agradables para un usuario. La estructura de la propuesta domótica se ha realizado en forma modular, para facilitar el análisis, la revisión y la reestructuración del sistema.

**Palabras clave:** Sistema domótico, HVAC, control difuso, interfaces.

### **Development of a Home Automation System Using Fuzzy Controllers, and Manual Controllers, Implemented in LabView and Arduino IDE**

**Abstract.** In this work, the design, development, and implementation aspects of a home automation system are addressed, this scheme has been developed by means of the LabView and Arduino IDE development tools. The system seeks to manage common aspects within a home through fuzzy, and manual controllers, in order to the designed building can have an automatic response, but also it can be controlled at the user's convenience at situations different from everyday ones. The security of the building is considered through the implementation of an

interface which can monitor the building through an IP camera. This work was carried out, seeking to generate comfortable, and pleasant environments for the user. The structure of the house automation proposal has been made in a modular way, to facilitate the analysis, revision and restructuring of the system.

**Keywords:** Home automation system, HVAC, fuzzy control.

## **1. Introducción**

El ser humano siempre ha buscado la forma de modificar su entorno, estas modificaciones son realizadas con la intención de generar un mayor grado de comodidad al existente. Las modificaciones incluyen, la generación de casas, o espacios donde descansar y realizar sus actividades de ocio y personales.

El avance de la tecnología y el uso de internet, puede permitir que una vivienda u edificio pueda ser modificado de tal manera que un usuario pueda olvidarse de ciertas tareas rutinarias y permitir que la casa en sí misma, pueda realizar dichas tareas, tomando decisiones “propias”. Este tipo de hábitats pueden ser considerados como casas inteligentes o sistemas domóticos.

Según la Asociación Española de Domótica e Inmótica, “un sistema domótico debe ser capaz de recoger información proveniente de sensores, procesarla y emitir órdenes” [1].

La domótica genera una integración dentro de un edificio u hogar de diferentes áreas del conocimiento, tales como las telecomunicaciones, la informática, la electrónica, y la inteligencia artificial. Estas estructuras son diseñadas con la finalidad de mejorar la calidad de vida de los usuarios, pero también se realizan buscando mejorar el ahorro energético y la protección al medio ambiente.

Un sistema domótico (SD) se compone de dos elementos, software y hardware. El hardware lo componen todos los dispositivos que se encargan de recibir señales del entorno, y efectuar cambios al entorno, como son los sensores y actuadores. El software por otro lado, es el responsable de analizar y procesar las señales de entrada, y gestionar las señales de salida.

Principalmente, se pueden distinguir dos tipos de arquitecturas de estos entes tecnológicos: la arquitectura centralizada, y la arquitectura distribuida.

En un sistema con arquitectura distribuida, cada dispositivo tiene un pequeño procesador propio que gestiona la información que se le ha sido pre programada por el fabricante, generando aplicaciones para funciones específicas. Este tipo de arquitectura, tiene la ventaja de que cada dispositivo cuenta con un alto nivel de autonomía, pero es debido a esta característica, que no se puede obtener una gran potencia del sistema, ya que todo el potencial está dividido en partes pequeñas.

Por otro lado, en la arquitectura centralizada, se tiene solamente un controlador, que es el encargado de recibir, procesar, y gestionar todas las señales de entrada y salida de un sistema. Al contrario de un sistema distribuido, su principal ventaja es el potencial de inteligencia que el sistema puede tener, sin embargo, si el procesador principal falla, todo el sistema falla.

Estas estructuras se han presentado como el futuro de las viviendas y edificios, buscando que dichos inmuebles puedan conectarse directamente a internet, gestionando aspectos tales como las compras de consumibles y alimentos; buscando que se pueda regular el uso del consumo energético y el cuidado al medio ambiente; y obviamente, buscando que los usuarios se olviden de tareas rutinarias, y se genere un estado de confort y tranquilidad.

Es debido a las circunstancias mencionadas, que es de gran importancia tecnológica y comercial el desarrollo de sistemas domóticos que puedan ser implementados y modificados de una manera sencilla y económica.

En este proyecto se presenta un SD que ha sido diseñado con la finalidad de que pueda ser manejado, analizado y reestructurado de una manera sencilla. Esta característica estará determinada por el hecho de que el sistema se ha planteado de forma modular, donde cada módulo representara una de las funciones del diseño domótico, pudiéndose reestructurar (en caso de falla o mal funcionamiento) de manera individual, sin tener que alterar ningún otro modulo del sistema. La estructura planteada, gestiona aspectos tales como la temperatura, luminosidad y monitoreo de los interiores de un edificio, y por medio del control de estos aspectos, se busca generar un entorno agradable, fresco y seguro, sin importar los cambios climatológicos que puedan ocurrir en el exterior.

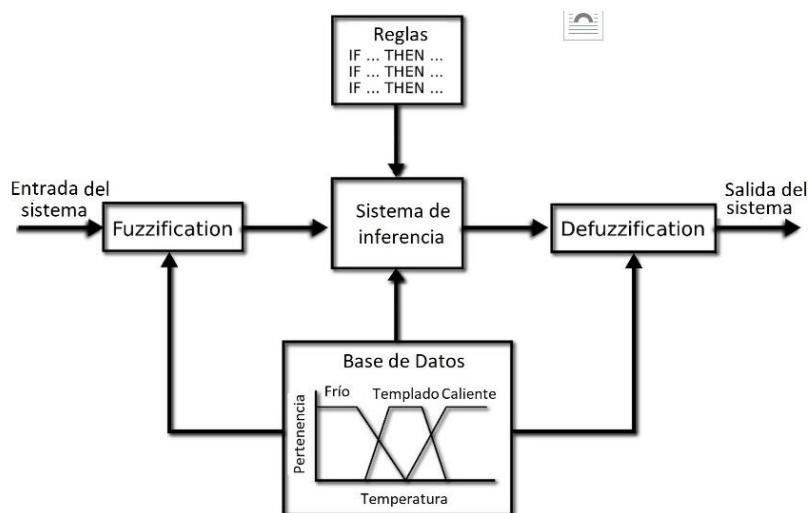
## **2. Preliminares**

### **a. Lógica difusa**

La lógica difusa surge entre los años 60's y 70's propuesta por *Lofti A Zadeh* [2]. Esta lógica permite simular los mecanismos de razonamiento humano para el control de sistemas, basados en la experiencia, proporciona un modelo matemático con el que se puede tratar la incertidumbre de los procesos cognitivos humanos y de este modo poder resolver problemas usando métodos matemáticos y computacionales.

Básicamente la lógica difusa es una forma de representar matemáticamente la incertidumbre y la vaguedad de un problema. Al buscar la solución del problema que fuere, se cuenta con variables de entrada y variables de salida, la lógica difusa permite la transformación entre las variables de entrada a su correspondiente salida privilegiando las características de significado en vez de las de precisión.

La lógica difusa emplea valores de números reales comprendidos entre 0 y 1 para indicar si un hecho es totalmente falso (0) o totalmente cierto (1), esto es el grado de pertenencia de un valor de entrada en un conjunto difuso, al ser multivaluada, una entrada puede pertenecer a más de un conjunto difuso al mismo tiempo, con un distinto valor de pertenencia para cada uno. Los conjuntos difusos, de este modo proporcionan una transición suave entre una característica y otra. Una vez que se han determinado los distintos valores de pertenencia de una entrada a cada uno de los conjuntos en el universo de discurso se dice que la variable se ha *borrosificado*.



**Fig. 1.** Esquema básico de un sistema de control difuso.

A partir de la lógica difusa, se han generado los denominados controladores difusos, sistemas que han sido eje principal de desarrollo de una gran cantidad de aplicaciones e interfaces [3,4]. Un sistema de control estándar basado en lógica difusa se puede representar de acuerdo a la figura 1.

## b. Arduino

Según la página oficial del proyecto Arduino (<https://www.arduino.cc/>), Arduino es una plataforma electrónica de código abierto basada en hardware y software de fácil manejo. Las placas Arduino pueden leer entradas por medio de sensores y botoneras, procesar estas entradas, y generar controles de diferentes tipos de actuadores en sus salidas.

Esta tarjeta de operaciones, se ha convertido en el proceso central de miles de proyectos, que van de proyectos muy simples a proyectos científicos sumamente complejos.

Arduino fue desarrollado en el Ivrea Interaction Design Institute como una herramienta que pudiera ser usada fácilmente por cualquier persona sin experiencia en electrónica y programación para la generación de prototipos tecnológicos. Existen varias versiones de la placa Arduino, sin embargo, la placa utilizada en este proyecto ha sido la placa UNO.



### **3. Estado del arte**

Muchos trabajos de implementaciones y propuestas domóticas se han presentado en la literatura científica, tratando principalmente temas relacionados al confort dentro de un edificio y al cuidado ambiental, dejando a un lado la facilidad de desarrollo, trabajos que en muchos casos utilizan la lógica difusa para su implementación. En esta sección solamente mencionaremos algunos de ellos y sus principales características, para dar un panorama general del desarrollo de este tipo de sistemas.

Existen implementaciones domóticas muy interesantes desde un punto de vista científico, por ejemplo, Pardo, Strack y Martínez en [5] presentaron un sistema para controlar dispositivos domésticos utilizando servicios web y una conexión doméstica común a Internet. Ellos mencionan que los dispositivos se pueden conectar a una computadora en el hogar y se pueden manipular o programar local o remotamente accediendo a un servidor web, lo que permite la independencia entre la aplicación local que controla los dispositivos desde el hogar y el servidor que permite el acceso remoto.

En [6], Rodríguez, Piedra e Iribarne, realizaron un sistema de aprendizaje que recopila información de la interacción del usuario con el sistema, y genera las reglas correspondientes que definirán el conocimiento del entorno domótico. Los autores mencionan que en su trabajo, ellos agregan la novedad de que las reglas que definirán el proceso de adaptación no son estáticas y preestablecidas, sino que pueden modificarse a lo largo del tiempo.

Una propuesta por demás interesante, es la presentada por Ché y Pardons, en [7], donde se menciona que el objetivo en una casa inteligente es aprender, reconocer y automatizar los patrones de interacción entre un individuo y los diversos dispositivos de automatización del hogar. Los autores consideran este ejemplo: el patrón es despertarse, apagar el despertador y luego hacer café. Un sistema de aprendizaje inteligente podría aprender este patrón y automáticamente preparar café por la mañana, cuando la persona apaga el despertador.

Como ya se mencionó, la generación de sistemas más amigables con el ambiente es uno de los principales objetivos de los sistemas domóticos, tal como lo muestran Villar, De La Cal y Sedano en [8]. Ellos presentan una solución de sistema multi agente para la reducción del consumo de energía en sistemas de calefacción de casas. En su propuesta se tiene una unidad central de control (CCU) responsable de minimizar el consumo de energía que interactúa con los calentadores. La CCU incluye un modelo difuso y un controlador difuso, y hace uso del concepto de balance de energía para distribuir la energía entre los calentadores.

Los proyectos domóticos, no solo buscan cuidar aspectos ambientales y de confort, también buscan generar espacios que puedan permitir el mejoramiento de la calidad de vida [9].

Por otro lado, Arduino ha jugado un papel importante en el caso de este tipo de sistemas, comportándose como parte principal en muchos de ellos. Por ejemplo, en [10] se diseñó e implementó una casa inteligente remotamente controlable, energéticamente eficiente y altamente escalable con características básicas que protegen la comodidad y la seguridad de los residentes. En este caso, los autores realizaron una red doméstica, por medio de sensores y actuadores. En su proyecto, la placa Arduino funcionó como

procesador central de la red. Arduino también sirvió como punto de comunicación entre el sistema diseñado y una aplicación *Android*. De manera similar, en [11] fue mostrado un sistema de control de hogar y control ambiental de bajo costo y flexible. Los autores emplearon un micro servidor web integrado en el microcontrolador Arduino Mega 2560, con conectividad IP para acceder y controlar dispositivos de forma remota.

Chandramohan y otros autores en [12] presentaron un sistema de monitoreo y control hogareño económico y flexible con la ayuda de un servidor micro-web integrado con conectividad de protocolo de internet (IP) para acceder y controlar equipos y dispositivos remotamente, usando una aplicación de teléfono inteligente basada en Android. En este proyecto, la placa Arduino funciona como operador principal de todo el sistema, recibiendo la información de los sensores y generando la comunicación con el dispositivo Android definido.

Como se ha podido leer, la placa Arduino ha sido pieza importante en diversos sistemas domóticos, sin embargo, en los sistemas mencionados la interfaz está definida por medio de una aplicación en el sistema operativo Android, caso contrario a la propuesta presentada en este trabajo, donde la interfaz de usuario estará gestionada por LabView.

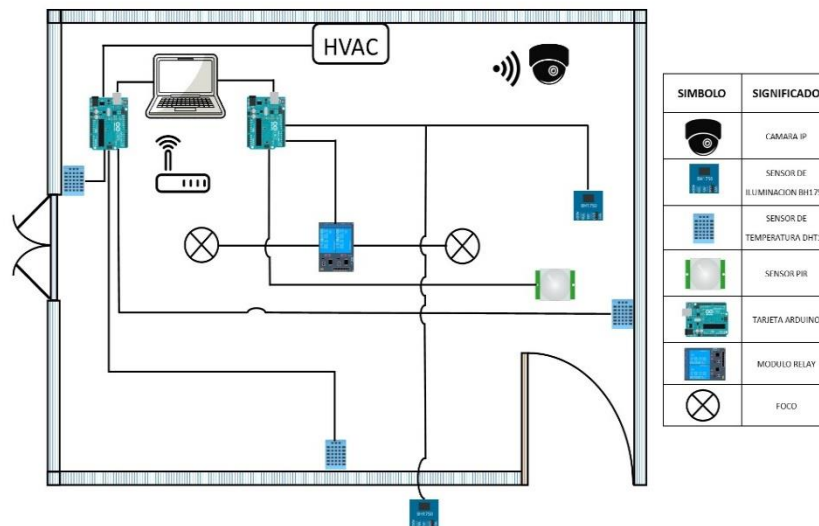
## 4. Desarrollo del proyecto

En esta sección se describe el proceso de desarrollo del proyecto, el sistema difuso utilizado, y la generación de la interfaz desarrollada por medio de *LabView* y sus instrumentos virtuales (VI) para generar el control de nuestro SD.

El SD está desarrollado en función de un sistema tipo HVAC (*Heating, Ventilating and Air Conditioning*), un controlador difuso, controladores on/off, controladores de luminosidad, así como un sistema de cámara IP. La plataforma en la cual se desarrolló el sistema completo es LabVIEW de *National Instruments* NI, obteniendo información de la placa Arduino. En esta sección, se describirá cada uno de los elementos que conforman al SD presentado.

### a. Esquema general

El sistema se ha planteado como un proceso modular, en donde cada módulo es independiente de los demás en su análisis, implementación, diagnóstico y reestructuración, sin embargo, todos estos módulos son monitorizados y pueden ser controlados manualmente por medio de una interfaz realizada en LabView. El sistema propuesto consta de una serie de tarjetas de desarrollo Arduino que leerán la información de los sensores y enviarán las señales de control a los elementos de salida. La comunicación entre las tarjetas y la computadora que ejecuta LabView, se realiza por medio del puerto serial. En el esquema se tienen los valores de entrada: sensor de temperatura, sensor de humedad, sensor de luz, sensor de presencia, código de acceso vía Wi-Fi para el inicio de la cámara IP y los botones virtuales para controlar los focos en forma On-Off. Como elementos de salida se tiene un ventilador (controlado por medio de un sistema difuso), un calefactor (controlado por medio de un sistema difuso),



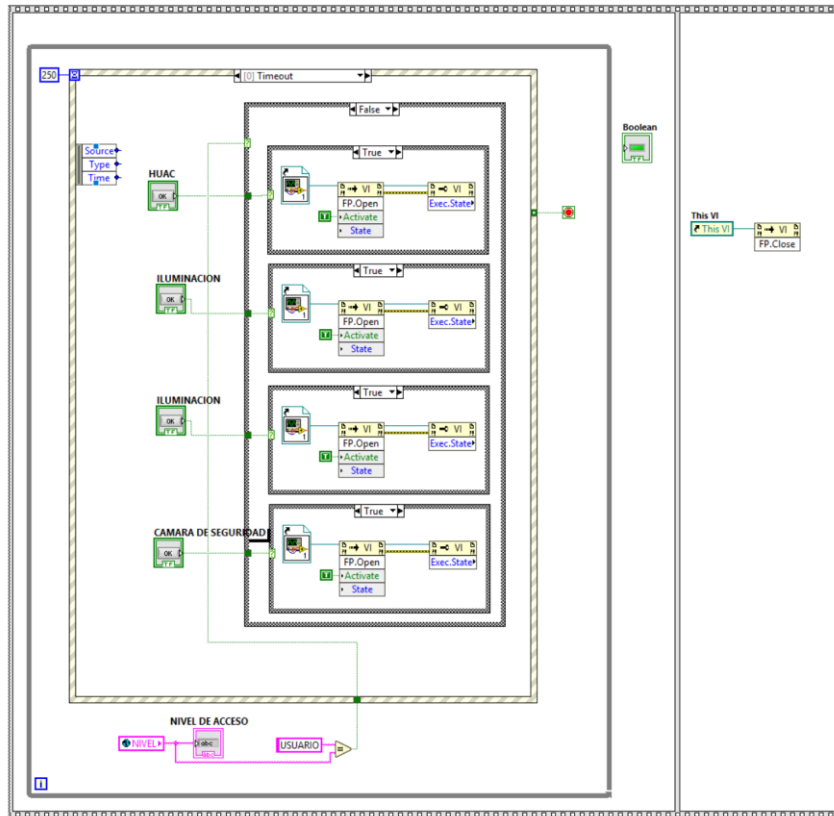
**Fig. 2.** Esquema general de conexiones de entradas y salidas del sistema propuesto, donde HVAC representa al ventilador y al calentador.

las señales que las luces recibirán, y la activación de la cámara IP. Todos estos dispositivos fueron implementados en una maqueta, la estructura del diseño propuesto de la maqueta se puede ver en la figura 2. El esquema de software que engloba todo el procesamiento de los datos, puede observarse en la figura 3.

#### b. Selección de herramientas a utilizar

Existen en el mercado una gran cantidad de tarjetas de desarrollo que puede ser utilizadas para el diseño de prototipos y proyectos tecnológicos, entre ellas están Raspberry Pi (<https://www.raspberrypi.org/>), Beagle Bone (<http://beagleboard.org/bone>) y Arduino. Cada una con sus características particulares, estas características pueden observarse en la tabla 1.

En la tabla 1, podemos observar que tarjetas como Raspberry Pi y BeagleBone, tienen mayores tamaños de memoria, y son mucho más rápidas que un Arduino UNO, sin embargo, estas tarjetas están principalmente enfocadas a software, a diferencia de la tarjeta Arduino que se enfoca a hardware. Por este motivo, esta tarjeta está más cercana a las necesidades que se tienen para el proyecto planteado, ya que lo que se busca es la interacción de elementos de hardware con la interfaz de LabView. La tarjeta Arduino es la tarjeta más económica en el mercado, lo que la hace accesible a casi cualquier sector poblacional. A pesar de las ventajas ya mencionadas, otra circunstancia por la que se ha elegido a Arduino sobre las otras tarjetas, es por la compatibilidad que esta tarjeta tiene con prácticamente cualquier sistema, incluyendo una compatibilidad directa con LabView.



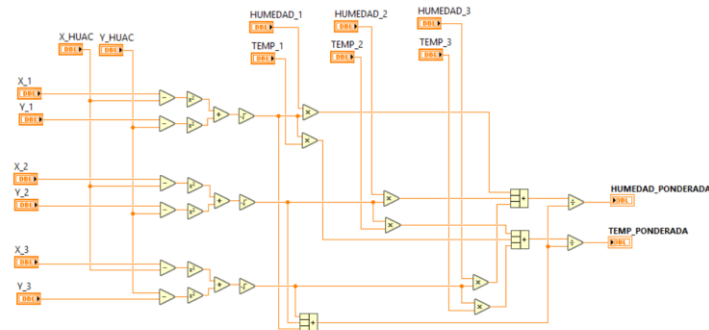
**Fig. 3.** Esquema general del software que procesa y gestiona las señales de entrada y salida del SD propuesto.

Esta situación permite que se pueda utilizar este tipo de tarjeta bajo distintos esquemas, situación que no presentan de manera directa las demás tarjetas de desarrollo. Principalmente se ha usado Arduino por su facilidad de reconfiguración.

### c. Control de iluminación

Este control está enfocado a generar estancias que tengan un buen nivel de iluminación, iluminación que permitirá a los usuarios del edificio, o casa, sentirse cómodos bajo estas circunstancias lumínicas. Al generar un control inteligente de encendido de luces, se logra que este ambiente agradable, se pueda mantener sin importar el horario o las circunstancias existentes en el entorno.

La gestión del control de iluminación dentro del sistema planteado se realiza mediante un controlador que toma como variables principales los datos obtenidos de sensores BH1750 (sensor de luz) dentro y fuera de la habitación, el uso principal que tiene la habitación, y la presencia de alguien en la habitación.



**Fig. 5.** Ponderación de las mediciones obtenidas por medio de los sensores.

**Tabla 2.** Reglas de evaluación para las entradas de temperatura y humedad, y su evaluación para la salida controlada por niveles para el ventilador.

HUMEDAD			
	BAJA	MEDIA	ALTA
TEMPERATURA			
	MUY FRIO	N0	N0
	FRIO	N0	N1
	TEMPLADO	N0	N2
	CALIENTE	N2	N4
	MUY CALIENTE	N3	N5

El controlador desarrollado en LabView toma estos 3 valores por medio de una tarjeta Arduino, los evalúa y define la cantidad lumínica apropiada para la habitación, es obvio mencionar que si, por ejemplo, no hay luz en la habitación, pero tampoco existe una presencia en ella, el controlador definirá que no es necesario encender los focos de dicha habitación. El diagrama de bloques que muestra este controlador se observa en la Fig. 4.

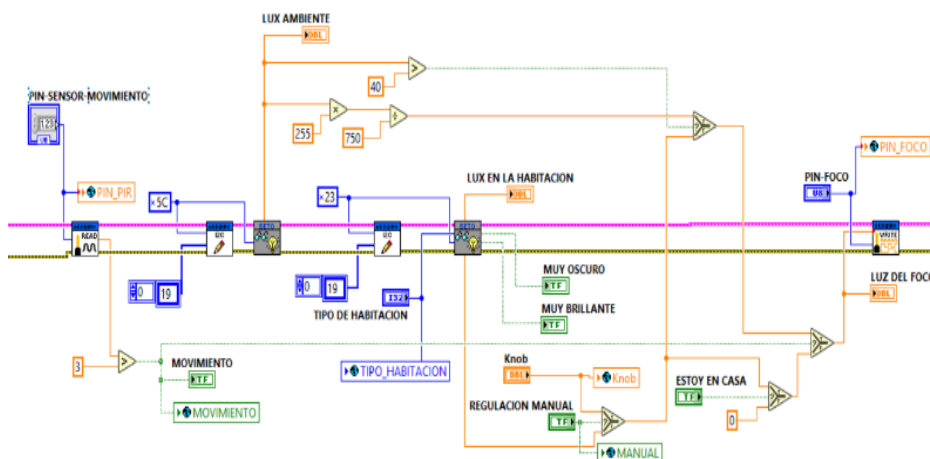
#### d. Control de temperatura

El sistema de control de temperatura está enfocado principalmente a brindar confort de los usuarios, manteniendo una temperatura adecuada en cada momento para las personas que se encuentren dentro del inmueble domótico, controlando también, las temperaturas en zonas particulares del edificio.

El Instituto para la Diversificación y Ahorro de la Energía (IDAE) indica que la temperatura ideal para sentir confort va de los 19 a los 21 grados en una habitación o casa, sin embargo, también mencionan que una temperatura de 21°C es suficiente para

**Tabla 1.** Principales características de distintas tarjetas de desarrollo.

Características	Raspberry Pi 3	BeagleBone	Arduino UNO
Memoria	1.0 GB	512 MB	2 KB
Velocidad de procesamiento	1.2 Ghz	1.0 GHz	16 MHz
Pines de entrada-salida	40	92	14
Puertos USB	2	1	1
Precio	53 USD	85 USD	20 USD



**Fig. 4.** Diagrama principal para la gestión de iluminación.

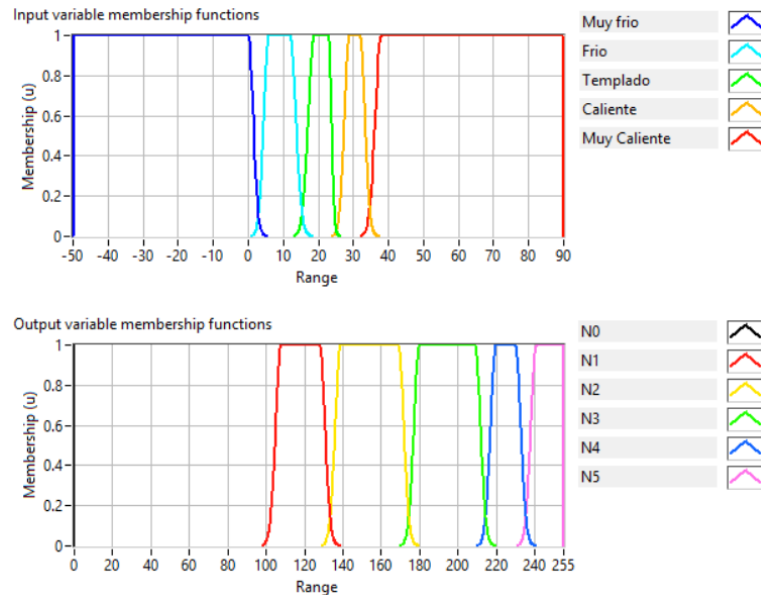
mantener el confort de una vivienda [2], aunque dependerá también de la cantidad de gente que se encuentre en ella.

Para el control de temperatura se utilizó la funcionalidad de un sistema HVAC. Una placa Arduino se encarga de recolectar los datos de los sensores de temperatura y humedad. Estos datos son enviados a un controlador difuso diseñado en LabView. El sistema difuso regresa los valores de respuesta hacía dos salidas: el encendido en diferentes potencias de un ventilador, y el apagado y encendido de un calefactor.

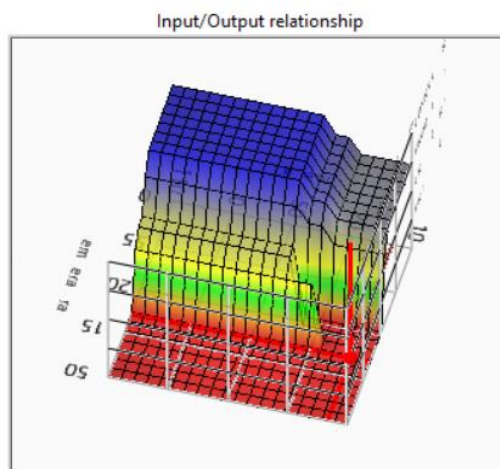
El sistema para el control de la temperatura consta de varias partes entre las principales se encuentran: Lectura de sensores, ponderación de mediciones y controladores difusos.

La ponderación es algo importante en un sistema HVAC ya que se toma en cuenta la posición del sensor con respecto al sistema de ventilación para así darle un valor de importancia más certero al momento de establecer la temperatura y humedad del lugar. El cálculo de la ponderación necesaria para cada uno de los sensores puede verse en la figura 5.

En este módulo se utilizó un etiquetado (generación de conjuntos difusos) para el manejo de los niveles de temperatura y humedad, el etiquetado y acciones ante ciertos niveles se manejaron mediante controladores difusos, los cuales nos facilitan la toma de decisiones y brindan una mayor certidumbre al usuario. El controlador fue diseñado con la herramienta Fuzzy System Designer que ofrece LabView.



**Fig. 6.** Conjuntos difusos diseñadas para las variables de entrada (temperatura y humedad) y de salida (PWM y calefactor).



**Fig. 7.** Comportamiento general del sistema difuso, donde se pueden simular diferentes valores de entrada de humedad y temperatura.

Las variables de entrada a considerar para este controlador son la temperatura y la humedad, mientras que en las variables de salida se tiene el PWM (Pulse Width Modulation) que controla la potencia del ventilador; y también se considera el encendido y apagado de un calefactor como salida 2. Los conjuntos difusos generados para el control del ventilador y del calefactor pueden verse en la figura 6.



Fig. 8. Interfaz de login.



Fig. 9. Panel de control de aplicaciones del sistema.

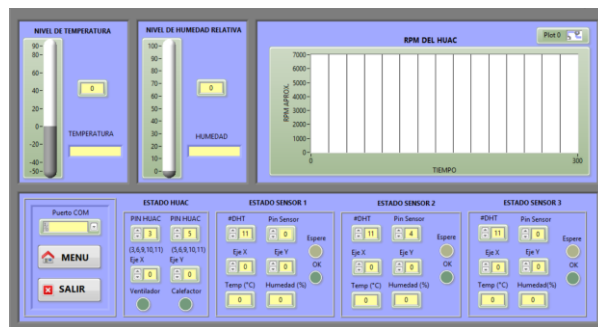


Fig. 10. Panel de administrador del módulo HVAC.



Fig. 11. Panel de usuario del módulo de luminosidad.

El establecimiento de las reglas para la evaluación de las variables de entrada, se realizaron tomando en cuenta los efectos que tienen los niveles de humedad en un ambiente, y con esto establecer la velocidad con la que funciona el ventilador y de igual forma si se enciende o mantiene apagado el calefactor. En la tabla 2 se observa el conjunto de reglas que rigen el controlador difuso.

La herramienta de diseño difuso de LabView, nos permite visualizar el comportamiento general del funcionamiento del controlador, y ver su respuesta ante



valores de entrada dados por el sensor de humedad y de temperatura. Dicho comportamiento puede verse en la fig. 7.

#### **e. Video vigilancia**

La idea general de un sistema de cámaras remoto en el caso de la domótica, es permitir al usuario monitorear a larga distancia los cambios o eventualidades que sucedan tanto al interior como al exterior de un hogar, tal que solo una persona autorizada pueda acceder y observar a través de dichas cámaras desde un punto remoto.

Este módulo dentro del SD, tiene como objetivo el poder generar un sistema de seguridad utilizando cámaras IP, con la intención de brindar no solo comodidad sino también seguridad de forma remota al esquema propuesto.

Para la implementación del sistema se utilizó la cámara de un *Smartphone*, lo que es un ejemplo de la ganancia económica que puede ofrecer el sistema, al no necesitar utilizar dispositivos especiales.

#### **f. Interfaz del sistema domótico**

La interfaz del SD, se realizó considerando un ambiente amable con el usuario y fácil de utilizar. Esta interfaz contiene una etapa de acceso, cuando se inicia la aplicación el sistema manda una interfaz de *login* en el cual se ingresan un nombre de usuario y contraseña para la identificación de usuarios y administradores, esto tiene como finalidad el establecer un nivel de acceso. El ingreso al sistema puede verse en la figura 8.

En caso de que tanto el usuario como la contraseña estén registrados en la base de datos del sistema, se mostrara la ventana de la figura 9 donde se pueden elegir las aplicaciones a las que se desea ingresar y también se mostrara el nivel de acceso.

En caso ingresar al módulo del HVAC (control de temperatura) en modo de administrador, se mostrará la interfaz de la Fig. 10. En este módulo podemos controlar la asignación de los puertos de los sensores y el ventilador. Al iniciar la aplicación, la interfaz nos arrojará las temperaturas que se obtengan de los sensores, la humedad relativa y las revoluciones del motor del ventilador, todo esto gestionado por medio del controlador difuso.

En caso de ingresar como usuario, solamente se permite observar la temperatura, humedad relativa, la activación del ventilador y que sensores se encuentran realizando mediciones.

Cuando se ingresa como administrador en el módulo de iluminación, se despliega la pantalla de la Fig. 11, en este módulo se conjuntan el controlador On/Off y el control de luminosidad. En la aplicación podemos controlar el encendido y apagado de los focos, así como la asignación de los puertos donde se controla cada uno, podemos controlar la iluminación de manera manual y observar la intensidad de luz que emiten los focos, se puede elegir el tipo de habitación en la cual están colocados los sensores, ver la iluminación de la habitación y observar si existe alguna presencia dentro de ella. En el acceso de usuario, solamente se permite usar el controlador On/Off de los focos,

observar los datos obtenidos por los sensores de luminosidad y controlar la cantidad de luz que emiten los focos con la regulación manual.

Por último, si se ingresa a la aplicación de cámara de seguridad la aplicación abrirá una pantalla, en donde se muestra el video que se recibe a través de un dispositivo móvil, el sistema permite monitorear una cámara que esté conectada a la misma red que la computadora donde se esté corriendo la aplicación del SD.

#### **g. Evaluación del sistema propuesto**

Como ya se ha mencionado, se generó un sistema en LabView que obtiene la información proveniente de los Arduinos por comunicación serie, procesa la información, calcula las salidas, y envía estos datos de regreso a las placas Arduino. El sistema físico se implementó en una maqueta, donde fueron instalados todos los sensores y elementos de salida ya descritos.

La maqueta física fue sometida a diferentes cambios de temperatura y humedad, cambios generados por calentadores, encendedores, y atomizadores de agua. El programa realizado en LabView, muestra en tiempo real los valores de salida que el sistema difuso va calculando, con estos valores en pantalla es fácil observar, la velocidad con la que el ventilador trabaja, y los momentos en los que el calentador es activado o desactivado. Las pruebas realizadas en este módulo, lograron generar ambientes confortables, estabilizando una temperatura promedio de 21°C dentro de todas las secciones de la maqueta.

El control lumínico del sistema se lleva a cabo en dos formas, por un lado, LabView recibe la información proveniente de los sensores de luz y de presencia, y en base a los valores obtenidos, realiza el cálculo para definir el mejor nivel de iluminación. Las pruebas realizadas mostraron que de esta manera se evitaba realizar un gasto innecesario de luz, cuando no hay una presencia en la habitación de la maqueta o cuando había suficiente luz externa dentro de las habitaciones, el sistema no genera emisión lumínica, sin embargo, a pesar del gasto energético que se puede ahorrar, el sistema cuenta con una interfaz que permite a un usuario autorizado encender las luces sin importar las demás variables a evaluar. Esta es una característica importante del sistema, ya que permite al usuario poder controlar el inmueble cuando las circunstancias salen de lo cotidiano.

Una de las principales ventajas de esta propuesta, es la posibilidad de reestructurar cualquier módulo de forma independiente, sin necesidad de alterar o siquiera tener que ingresar en cualquier otro modulo. Esta circunstancia permite la libertad de acción sobre cualquier modulo, sin considerar que pueda alterarse cualquier otro sub sistema. Si un módulo se estropea, solo se reestructura ese modulo, y el resto del SD permanece funcionando de manera normal.

Una de las principales cualidades de la estructura planteada, es la posibilidad de poder elegir entre los tipos de control automático y manual por medio de la interfaz en LabView. Esto permite que el usuario pueda ajustar el comportamiento de su sistema a su entero agrado y conveniencia. Si existe una circunstancia que salga del comportamiento usual del usuario tal como una enfermedad, él podrá solamente tomar el control del edificio y definir las mejores condiciones en ese instante.

El inmueble planteado, ha mostrado funcionar de manera adecuada a las pruebas realizadas, y como ventaja principal, este esquema puede diagnosticarse y modificarse de manera simple, debido a que se ha desarrollado como un conjunto de módulos.

## **5. Conclusiones**

En la búsqueda de un nivel de automatización óptimo en los hogares, oficinas, edificios y/o lugares públicos es necesario integrar diversas áreas de la ingeniería, física y de la salud para lograr desarrollar un sistema que brinde la comodidad adecuada en nuestro entorno, y a la vez la seguridad de que el sistema está trabajando "pensando" en las necesidades del usuario.

En este trabajo, se ha presentado una opción de ambiente domótico que es una opción económica en hardware dentro del ámbito de la automatización de viviendas. En este proyecto, se han utilizado componentes y herramientas conocidas como puntos de soporte, tal como la placa de desarrollo Arduino y la plataforma LabView.

Con la intención de facilitar el desarrollo y manejo interno del proyecto, se planteó una estructura e implementación en una forma modular, por lo cual cada módulo puede ser añadido, eliminado o modificado sin afectar a todo el sistema en general. Esta es una de las principales ventajas de la propuesta, ya que, si una sección del sistema falla, esa falla no se propagará a ninguna otra sección, permitiendo identificar y solucionar rápidamente el problema.

Este sistema fue planteado utilizando una combinación de diferentes formas de control, sin embargo, la utilización del controlador difuso, permite generar un ambiente completamente independiente y que responde a las necesidades para las que fue diseñado, es decir, permite dar comodidad a un usuario sin que él se dé cuenta de que la climatización del edificio o vivienda se está modificando constantemente, obviamente de manera precisa a sus necesidades.

Por otro lado, el esquema planteado permite poder modificar el comportamiento de los elementos de salida de manera manual, esta es otra gran ventaja que le permite al usuario tener un control total de su ambiente, aun cuando situaciones diferentes a las cotidianas sucedan.

Para este sistema también se generó una interfaz que es amable con el usuario y que permite facilidad de entendimiento, esta interfaz se dividió en dos tipos de acceso, el de administrador y el de usuario común, para que el acceso al sistema tenga jerarquía y se pueda mantener un control adecuado del funcionamiento del sistema.

**Agradecimientos.** Este trabajo fue en parte financiado por el proyecto PAPIIT IN 113316, y el proyecto PIAPI 1634 de la UNAM.

## **Referencias**

1. Asociación Española de Domótica e Inmótica: ¿Qué es domótica? (2017)
2. Zadeh, L.A.: Fuzzy sets. Information and control, 8(3), pp. 338–353 (1965)

3. Pires, G., Nunes, U.: A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. *Journal of Intelligent & Robotic Systems*, 34(3), pp. 301–314 (2002)
4. Ye, C., Yung, N.H., Wang, D.: A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(1), pp. 17–27 (2003)
5. Pardo, M.E., Strack, G., Martínez, D.C.: A domotic system with remote access based on web services. *Journal of Computer Science & Technology*, 8 (2008)
6. Rodríguez-García, D., Piedra-Fernández, J. A., Iribarne, L.: Adaptive Domotic System in Green Buildings. In *Advanced Applied Informatics (IIAI-AAI), IIAI 4th International Congress on, IEEE*. pp. 593–598 (2015)
7. Ché, N.K., Pardons, N., Vanrompay, Y., Preuveneers, D., Berbers, Y.: An intelligent domotics system to automate user actions. In: *Ambient Intelligence and Future Trends–International Symposium on Ambient Intelligence (ISAmI’10)* pp. 201–204, Springer, Berlin (2010)
8. Villar, J.R., De La Cal, E., Sedano, J.: Energy saving by means of fuzzy systems. In: *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, pp. 155–167 (2007)
9. Vacher, M., Istrate, D., Portet, F., Joubert, T., Chevalier, T., Smidtas, S., Méniard, S.: The sweet-home project: Audio technology in smart homes to improve well-being and reliance. In: *Engineering in Medicine and Biology Society, (EMBC’11) Annual International Conference of the IEEE*, pp. 5291–5294 (2011)
10. Baraka, K., Ghobril, M., Malek, S., Kanj, R., Kayssi, A.: Low cost arduino/android-based energy-efficient home automation system with smart task scheduling. In: *Computational Intelligence, Communication Systems and Networks (CICSyN’13) Fifth International Conference on. IEEE*, pp. 296–301 (2013)
11. David, N., Chima, A., Ugochukwu, A., Obinna, E.: Design of a home automation system using arduino. *International Journal of Scientific & Engineering Research*, 6(6), pp.795–801 (2015)
12. Chandramohan, J., Nagarajan, R., Satheeshkumar, K., Ajithkumar, N., Gopinath, P.A., Ranjithkumar, S.: Intelligent smart home automation and security system using Arduino and Wi-fi. *International Journal of Engineering and Computer Science*, 6(3) (2017)

# TFL<sup>PL</sup>: Programación con lógica de términos

J. Martín Castro-Manzano, L. Ignacio Lozano-Cobos

Facultad de Filosofía y Humanidades, UPAEP, Puebla,  
México

josemartin.castro@upaep.mx

**Resumen.** A partir del sistema lógico *Term Functor Logic* (TFL) y del concepto de base de datos aristotélica, en esta contribución presentamos los avances de un lenguaje de programación lógica que hemos denominado *Term Functor Logic Programming Language* (TFL<sup>PL</sup>).

**Palabras clave:** Lógica de términos, silogística, base de datos aristotélica.

## TFL<sup>PL</sup>: Programming with Term Logic

**Abstract.** Given the system *Term Functor Logic* (TFL) and the concept of Aristotelian database, in this paper we present the development of a programming language we call *Term Functor Logic Programming Language* (TFL<sup>PL</sup>).

**Keywords:** Term logic, syllogistic, Aristotelian database.

### 1. Introducción

Bajo la influencia directa de [26,28,29,32,16], en otro lugar hemos ofrecido una versión de una silogística (relacional con cuantificadores no-clásicos) que trata con una amplia gama de patrones inferenciales de sentido común con las ventajas de un enfoque algebraico (el sistema TFL<sup>+</sup>); y bajo la influencia directa de [5,8,6,7] y del mismo sistema TFL<sup>+</sup>, en otro lugar hemos presentado un sistema de diagramas lineales para razonar visualmente (el sistema TFL<sup>⊕</sup>) (cf. [19]). Ahora, bajo la influencia de estos dos últimos sistemas y la noción de base de datos aristotélica [17], en este trabajo presentamos los avances de un lenguaje de programación lógica que hemos denominado *Term Functor Logic Programming Language* (TFL<sup>PL</sup>).

Así pues, en esta contribución tratamos de alcanzar dos metas: *i*) introducir los elementos básicos de un sistema lógico de términos capaz de lidiar con una amplia gama de patrones inferenciales de razonamiento ordinario (TFL<sup>+</sup>) y *ii*)

presentar los avances de un lenguaje de programación diseñado a partir de tal sistema ( $\text{TFL}^{\text{PL}}$ ). Para alcanzar estos resultados presentamos el sistema de términos TFL [26,28,29,5,6,7] a la luz de lo que hemos llamado “la visión heredada de la lógica” (§2) y el concepto de base de datos aristotélica [17] (§3): esto permitirá apreciar la relevancia del lenguaje que proponemos; posteriormente exponemos los avances del lenguaje  $\text{TFL}^{\text{PL}}$  mediante una presentación de su sintaxis y su funcionamiento global (§4); al final, cerramos con algunas observaciones sobre trabajo futuro y problemas por resolver (§5).

## 2. La visión heredada de la lógica

La lógica estudia la relación de inferencia y para llevar a cabo tal estudio es costumbre hacer uso de lenguajes de primer orden. Así, por ejemplo, la lógica proposicional, la lógica de primer orden y la lógica de primer orden con identidad son sistemas lógicos definidos mediante lenguajes de primer orden:

$\{p, q, r, \dots, \neg, \Rightarrow\}$ ,  $\{a, b, c, \dots, x, y, z, \dots, f, g, h, \dots, A, B, C, \dots, \neg, \Rightarrow, \forall, \exists\}$   
 $\{a, b, c, \dots, x, y, z, \dots, f, g, h, \dots, A, B, C, \dots, \neg, \Rightarrow, \forall, \exists, =\}$ , respectivamente.

El origen de esta costumbre está relacionado con las ventajas de orden representativo que los lenguajes de primer orden ofrecen frente a sistemas más tradicionales. Russell, por ejemplo, popularizó la idea de que las limitaciones del programa lógico tradicional, i.e., silogístico (*vide* Apéndice A), se debían al análisis de las proposiciones en clave terminista, como triadas de términos sujeto y predicado unidos por una cópula [24]. Carnap generalizó esta consideración a toda la lógica tradicional al sostener que la única sintaxis disponible en este tipo de lógica es predicativa [4]. Ciertamente, las limitaciones de la sintaxis de términos ternaria (sujeto-cópula-predicado) generan dificultades para representar proposiciones singulares, relacionales y proposicionales, y si estos impedimentos parecen menores, consideremos un problema con consecuencias graves: la homogeneidad de términos. Geach argumenta:

*Our distinction between names and predicables enables us to clear up the confusion, going right back to Aristotle, as to whether there are genuine negative terms: predicables come in contradictory pairs, but names do not, and if names and predicables are both called “terms” there will be a natural hesitation over the question “Are there negative terms?” [10, p. 64]*

De acuerdo a este argumento, la homogeneidad de términos no permite preservar la distinción fundamental entre nombre y predicado. Esta incapacidad de la sintaxis de términos resulta problemática porque las funciones de un nombre y las propiedades de un predicado no son intercambiables: mientras la función de un nombre es nombrar, la de un predicado es predicar; pero predicar no es nombrar y nombrar no es predicar. Por tanto, como argumenta Geach, es lógicamente imposible un intercambio sintáctico entre los términos sujeto y

predicado sin un cambio en el sentido de las proposiciones, pues sólo un nombre puede ser un sujeto lógico, pero un nombre no puede mantener su rol de nombre si se convierte en predicado. Por ello, esta dificultad sintáctica es también una dificultad semántica que se antoja insalvable: entre una lógica de términos y la lógica genuina, dice Geach, sólo puede haber guerra [9, p. 54].

En contraste, la lógica genuina, caracterizada por la lógica de primer orden (LPO), sigue la sintaxis del paradigma Fregeano que resulta de abandonar el uso de términos para favorecer una gramática binaria de pares función-argumento. Estos pares promueven una sintaxis que incluye constantes ( $a, b, c, \dots$ ) o variables ( $x, y, z, \dots$ ) como argumentos para referir a objetos individuales como sujetos lógicos, además de relaciones ( $A, B, C, \dots$ ) como funciones para referir a conceptos, no a objetos, como predicados lógicos. Así, por ejemplo, una proposición como “Sócrates es mortal” no podría ser entendida como una relación entre términos sujeto y predicado, sino como un par función-argumento donde una constante, un elemento saturado y completo, digamos  $s$ , denota a un objeto de nombre “Sócrates” que funge como argumento de la función incompleta y no-saturada “... *es mortal*”, digamos  $Mx$ , de tal modo que  $Ms$  representa la proposición *Sócrates es mortal*: claramente, esta representación sintáctica no permite el intercambio de términos. Más todavía, dada esta sintaxis binaria, una proposición como “Todos los hombres son mortales” no puede ser entendida como una relación de términos sino como una relación entre variables y cuantificadores, digamos  $\forall x(Hx \Rightarrow Mx)$ , de tal modo que existe una diferencia sintáctica clara entre una proposición singular (como “Sócrates es mortal”) y una proposición universal (como “Todos los hombres son mortales”): esto será importante más adelante.

Así pues, de acuerdo a estas consideraciones, la lógica genuina sigue la sintaxis del paradigma Fregeano que resulta de abandonar el uso de una sintaxis ternaria (sujeto-cópula-predicado) para favorecer una sintaxis binaria (función-argumento). Esta sintaxis binaria promueve lenguajes de primer orden. Esta elección sintáctica es la que nos es familiar actualmente porque es la que nos acompaña en la docencia, la investigación y la aplicación de la lógica contemporánea: esta es la visión heredada de la lógica. Sin embargo, no hace falta mucho ojo crítico para notar que esta visión nos puede ser familiar, pero no por ello nos resulta natural. Woods comenta (el énfasis es nuestro):

*It is no secret that classical logic and its mainstream variants aren't much good for human inference as it actually plays out in the conditions of real life—in life on the ground, so to speak. It isn't surprising. Human reasoning is not what the modern orthodox logics were meant for. The logics of Frege and Whitehead & Russell were purpose-built for the pacification of philosophical perturbation in the foundations of mathematics, notably but not limited to the troubles occasioned by the paradox of sets in their application to transfinite arithmetic. [35, p. 404].*

Así pues, si bien la lógica genuina (clásica, según Woods) ha sido fundamental para el estudio de la inferencia en ciencias cognitivas e inteligencia artificial, no

deja de extrañarnos que, a pesar de su finalidad original, sea constantemente utilizada para modelar razonamiento en lenguaje natural. Consideremos, a este efecto, lo que hemos denominado “el reto de Bar-Hillel”:

*I challenge anybody here to show me a serious piece of argumentation in natural languages that has been successfully evaluated as to its validity with the help of formal logic. I regard this fact as one of the greatest scandals of human existence. Why has this happened? How did it come to be that logic which, at least in the views of some people 2,300 years ago, was supposed to deal with evaluation of argumentation in natural languages, has done a lot of extremely interesting and important things, but not this? [30, p. 256]*

Esto, ciertamente, es escandaloso. Sin embargo, desde finales de la década de los 60's Fred Sommers defendió una revisión y una revitalización de la sintaxis ternaria tradicional a la luz del reto de Bar-Hillel. Sommers, cercano a un proyecto de naturalización de la lógica, estaba preocupado por cómo es que razonamos. Así, por ejemplo, Sommers se preguntaba por las razones por las cuales un agente racional se da cuenta de que el siguiente par de creencias es inconsistente:

( $C_1$ ) Todo perro es animal pero ( $C_2$ ) alguien que quiere a un perro no quiere a un animal.

Por supuesto que LPO es capaz de ofrecer una respuesta al problema anterior haciendo uso de lenguajes de primer orden (sea  $C_1$ ,  $\forall x(Px \Rightarrow Ax)$ ; y  $C_2$ ,  $\exists x\exists y((Py \wedge Qxy) \wedge \forall z(Az \Rightarrow \neg Qxz))$ ) y métodos de demostración adecuados (Ecuación 1), pero como veremos más adelante, esta capacidad no necesariamente es relevante para comprender el razonamiento en lenguaje natural.

1	$\forall x(Px \Rightarrow Ax)$	$C_1$
2	$\exists x\exists y((Py \wedge Qxy) \wedge \forall z(Az \Rightarrow \neg Qxz))$	$C_2$
3	$\exists y(Py \wedge Qay) \wedge \forall z(Az \Rightarrow \neg Qaz)$	<b>E<math>\exists</math> 2, a/x</b>
4	$(Pb \wedge Qab) \wedge \forall z(Az \Rightarrow \neg Qaz)$	<b>E<math>\exists</math> 3, b/y</b>
5	$Pb \wedge Qab$	<b>E<math>\wedge</math> 4</b>
6	$\forall z(Az \Rightarrow \neg Qaz)$	<b>E<math>\wedge</math> 4</b>
7	$Ab \Rightarrow \neg Qab$	<b>E<math>\forall</math> 6, b/z</b>
8	$Pb \Rightarrow \neg Ab$	<b>E<math>\forall</math> 1, b/x</b>
9	$Pb$	<b>E<math>\wedge</math> 5</b>
10	$Qab$	<b>E<math>\wedge</math> 5</b>
11	$Ab$	<b>E<math>\Rightarrow</math> 8 y 9</b>
12	$\neg Qab$	<b>E<math>\Rightarrow</math> 7 y 11</b>
13	$Qab \wedge \neg Qab$	<b>I<math>\wedge</math> 10 y 12</b>
14	X	<b>EFSQ 4 a 13</b>
15	X	<b>EFSQ 3 a 14</b>

(1)



## 2.1. El sistema TFL

En este contexto, Sommers [26,28,29] y Englebretsen [5,6,7] desarrollaron una lógica más cercana a nuestro lenguaje natural. El resultado fue el sistema algebraico *Term Functor Logic* (TFL), el cual asume una sintaxis ternaria<sup>1</sup> y ofrece la siguiente gramática para representar proposiciones categóricas:<sup>2</sup>

- SaP :=  $-S + P = -S - (-P) = -(-P) - S = -(-P) - (+S)$
- SeP :=  $-S - P = -S - (+P) = -P - S = -P - (+S)$
- SiP :=  $+S + P = +S - (-P) = +P + S = +P - (-S)$
- SoP :=  $+S - P = +S - (+P) = +(-P) + S = +(-P) - (-S)$

Dada esta representación, TFL ofrece un método de decisión correcto, completo y simple para la silogística: una conclusión se sigue válidamente de un conjunto de premisas syss *i*) la suma de las premisas es algebraicamente igual a la conclusión y *ii*) el número de conclusiones con cantidad particular (*viz.*, cero o uno) es igual al número de premisas con cantidad particular [6, p.167]. Así, por ejemplo, si consideramos un silogismo válido, digamos un silogismo tipo **aaa-1**, podemos ver cómo la aplicación de este método produce la conclusión correcta (Tabla 1).

**Tabla 1.** Un razonamiento válido: **aaa-1**.

Proposición	TFL
1. Todo mamífero es animal.	$-M + A$
2. Todo perro es mamífero.	$-P + M$
3. Todo perro es animal.	$-P + A$

En el ejemplo de arriba podemos apreciar claramente cómo funciona el método: *i*) si sumamos las premisas obtenemos la expresión algebraica  $(-M + A) + (-P + M) = -M + A - P + M = -P + A$ , de tal suerte que la suma de las premisas es igual a la conclusión y la conclusión es  $-P + A$ , en lugar de  $+A - P$ , porque *ii*) el número de conclusiones con cantidad particular (cero en este caso) es igual al número de premisas con cantidad particular (cero en este caso).

Pero además, como Leibniz habría deseado [14], este sistema algebraico no sólo es capaz de modelar inferencias silogísticas, sino que es capaz de representar,

<sup>1</sup> Que podemos razonar sin elementos lingüísticos de primer orden—como variables individuales o cuantificadores—no es una idea novedosa (cf. [23,20,13]), pero el proyecto lógico de Sommers tiene un alcance más amplio: que sea posible usar una lógica de términos en lugar de un sistema de primer orden no tiene nada que ver con el hecho sintáctico, por decirlo de algún modo, de que podemos hacer inferencia sin cuantificadores o variables, sino con la visión más general de que el lenguaje natural es una fuente genuina de una lógica natural (cf. [27,15]).

<sup>2</sup> Aquí seguimos la presentación de [6].

en respuesta a las críticas mencionadas en la sección anterior, inferencias con proposiciones relacionales (Tabla 2), singulares<sup>3</sup> (Tabla 3) y proposicionales<sup>4</sup> (Tabla 4) con facilidad y claridad preservando la idea nuclear de TFL, a saber, que la inferencia es un proceso lógico que ocurre usando términos.

**Tabla 2.** Argumento relacional.

Proposición	TFL
1. Algunos caballos son más rápidos que algunos perros.	$+C + (+R + P)$
2. Los perros son más rápidos que algunos hombres.	$-P + (+R + H)$
3. La relación <i>más rápido que</i> es transitiva.	$-(+R + (+R + H)) + (+R + H)$
$\vdash$ Algunos caballos son más rápidos que algunos hombres.	$+C + (+R + H)$

**Tabla 3.** Argumento con singulares

Proposición	TFL
1. Todo hombre es mortal.	$-H + M$
2. Sócrates es hombre.	$-s + H$
$\vdash$ Sócrates es mortal.	$-s + M$

**Tabla 4.** Argumento proposicional

Proposición	TFL
1. Si $P$ entonces $Q$ .	$-[p] + [q]$
2. $P$ .	$[p]$
$\vdash Q$ .	$[q]$

Estos elementos básicos de TFL son suficientes para ofrecer una solución a los problemas de representación de las lógicas de términos que mencionábamos renglones arriba. Pero más todavía, estos elementos ofrecen razones para explicar, por ejemplo, por qué juzgamos instantáneamente que  $(C_1)$  *Todo perro es animal* y  $(C_2)$  *Alguien que quiere a un perro no quiere a un animal* son mutuamente inconsistentes. Consideremos, a este efecto, la derivación de la contradicción entre  $C_1$  y  $C_2$  en TFL, con fines comparativos (Tabla 5).

**Tabla 5.** Derivación de la contradicción entre  $C_1$  y  $C_2$  en TFL.

Proposición	TFL
$C_1$ . Todo perro es animal.	$-P_2 + A_2$
$C_2$ . Alguien que quiere a un perro no quiere a un animal.	$+(+Q_{12} + P_2) - (+Q_{12} + A_2)$
$\vdash$ Alguien que quiere a un animal no quiere a un animal.	$+(+Q_{12} + A_2) - (+Q_{12} + A_2)$

<sup>3</sup> Provisto que los términos singulares como *Sócrates* se representen con minúsculas.

<sup>4</sup> Dado que el razonamiento proposicional se puede representar de la siguiente manera,  $P := [p]$ ,  $Q := [q]$ ,  $\neg P := -[p]$ ,  $P \Rightarrow Q := -[p] + [q]$ ,  $P \wedge Q := +[p] + [q]$  y  $P \vee Q := - - [p] - - [q]$ , el método de decisión se comporta como resolución (cf. [21]).

Si comparamos la prueba de la contradicción de  $C_1$  y  $C_2$  en LPO (Ecuación 1) con la prueba de la contradicción en TFL (Tabla 5), podemos observar que en LPO no sólo necesitamos una traducción más compleja haciendo uso de variables y cuantificadores, sino también una demostración más larga para justificar que existe una contradicción; y sin embargo, a diferencia de TFL, tal demostración no ofrece luces sobre por qué *vemos* instantáneamente una inconsistencia entre  $C_1$  y  $C_2$ .

En contraste, la explicación en TFL tiene una naturalidad sintáctica y un conjunto de reglas simples para razonar. Y como LPO, por su origen, no posee estas características, no puede ofrecer información cognitivamente relevante sobre el modo en como razonamos. Y por ello, aunque LPO tiene mérito en la fundamentación de las matemáticas, no puede ser una lógica del razonamiento en lenguaje natural. Y como puede sospecharse en este punto, esta narrativa tiene ciertos nexos con la historia de los lenguajes de programación lógica [2].

### 3. Bases de datos aristotélicas

En efecto, de acuerdo con Mozes [17], el abandono de la lógica aristotélica, por parte de los computólogos, para favorecer LPO no está justificado (Cf. [12]). La lógica silogística es una lógica de términos que fue creada con el fin de comprender y guiar el razonamiento humano; pero como hemos comentado renglones arriba, esta no es la intención de LPO, y como resultado de esto, una lógica de términos no sólo tiene importancia cognitiva, sino también computacional.

En consecuencia, Mozes desarrolló el concepto de *base de datos aristotélica*. Una base de datos es aristotélica cuando posee las siguientes características:

- La habilidad de proveer explicaciones sobre las deducciones utilizadas en lenguaje natural.
- La habilidad de ofrecer información, en respuesta a preguntas dicotómicas (“sí/no”), si una versión más fuerte o débil de un “sí” puede ser probada.
- La habilidad de señalar resultados que no pueden ser probados pero que son posibles.
- La habilidad de sugerir reglas implícitas que si se añaden a la base de datos, podrían proveer respuestas afirmativas.
- La habilidad de indicar instancias en las cuales patrones no-deductivos, como analogías, pueden ser útiles.

Para obtener estas características, la estructura de una base de datos a la Mozes consiste de un conjunto de constantes para representar objetos y un conjunto de relaciones (no hay funciones) para representar propiedades de los objetos (volveremos a este punto en la siguiente sección pues, como puede notarse, Mozes asume una sintaxis Fregeana). La información sobre los objetos se expresa mediante hechos que consisten de relaciones aplicadas a objetos, por ejemplo, *hombre(Socrates)*. En este sentido, estas bases siguen una sintaxis similar a la de Prolog [31,3]. Una regla, por otro lado, consiste de un sujeto, que es la conjunción de una o más relaciones aplicadas a variables y constantes, y un

predicado, que es una relación única; más un tipo de regla que indica la conexión entre el sujeto y el predicado. Cuando se escribe una regla se escribe primero el predicado, luego el tipo de la regla y finalmente el sujeto. Hay cuatro posibles tipos de reglas que corresponden a las cuatro proposiciones categóricas (*vide* Apéndice A). Así, por ejemplo, `mortal(X) A hombre(X)` significa *Todo hombre es mortal* (en Prolog, `mortal(X) : -hombre(X).`).

Como en Prolog, en estas bases de datos existen dos tipos de consultas: consultas de respuesta, que especifican un hecho o una regla y tratan de probarlas; y consultas de recuperación, que regresan una respuesta dado el cumplimiento de una conjunción de relaciones. Después de obtener respuesta a una consulta, el usuario puede preguntar por una explicación de la respuesta. Si la consulta es de recuperación, el usuario puede especificar una constante y preguntar por qué dicha constante no fue regresada como respuesta. Si la consulta es dicotómica y no pudo ser probada como verdadera o falsa, el usuario puede preguntar por reglas implícitas. Además de probar conocimiento negativo de manera explícita, la base de datos también hace uso de la negación por fallo. Finalmente, los procesos deductivos de este tipo de base de datos se basan en la silogística. Consideremos, a modo de ejemplo, el siguiente fragmento adaptado de [17]:

```
\\Hechos
hombre(Socrates)
sabio(Socrates)
hombre(Joe)
edad(Joe,1)
hombre(John)
edad(John,15)
hombre(Peter)
edad(Peter,40)
perro(Fido)
obra_de_arte(MonaLisa)

\\Reglas
animal(X) A hombre(X)
animal(X) A perro(X)
mortal(X) A hombre(X)
mortal(X) E obra_de_arte(X)
hombre(X) E perro(X)
responsable(X) E ~hombre(X)
responsable(X) A sabio(X)
responsable(X) E bebe(X)
responsable(X) I adulto(X)
responsable(X) E idiota(X)
edad(X,0-120) A hombre(X)
bebe(X) A (hombre(X)^edad(X,0-2))
adulto(X) A (hombre(X)^edad(X,21-120))
```

```

idiota(X) I (hombre(X)^edad(X,30-50))

\\Ejemplo
> mortal(X) ^ ~responsable(X)
Joe
Por negacion por fallo: John
Posibilidad: Fido
> Explica Joe
Porque Joe es un hombre, es mortal.
Porque Joe es un hombre con edad 1, es bebe.
Porque Joe es un bebe, no es responsable.
> Explica Fido
Porque Fido es un perro, es animal.
Algun animal es mortal; por ejemplo, hombre.
Porque Fido es animal, tal vez es mortal.
Porque Fido es un perro, no es hombre.
Porque Fido no es hombre, no es responsable.
> Explica Socrates
Porque Socrates es sabio, es responsable.
> Explica Peter
Porque Peter es hombre con 40, es adulto.
Porque Peter es adulto, tal vez es responsable.

```

Las bases de datos aristotélicas, pues, usan reglas silogísticas como modelos adecuados de inferencia. La principal ventaja de este modo de hacer bases de datos es su cercanía cognitiva con el razonamiento en lenguaje natural. Esto sugiere, en opinión de Mozes, dos áreas en las que la lógica tradicional podría ser aplicada: aplicaciones en las cuales la interacción en lenguaje natural con humanos es ubicua; y aplicaciones que tienen como objetivo simular razonamiento humano cuando es preciso sugerir posibilidades o inducciones.

## 4. El lenguaje TFL<sup>PL</sup>

Ahora bien, si asumimos la meta de interacción en lenguaje natural con humanos, parece que es conveniente hacer uso de una base de datos aristotélica más poderosa en la medida en que sea capaz de representar un rango más amplio de inferencias de sentido común. Para lograr esto, a continuación hacemos una breve introducción del sistema TFL<sup>+</sup>.

### 4.1. El sistema TFL<sup>+</sup>

Peterson [22] y Thompson [32] desarrollaron extensiones para la silogística añadiendo cuantificadores adicionales: “la mayoría” (para proposiciones mayoritarias), “muchos” (para proposiciones comunes) y “poco” (para proposiciones

predominantes)<sup>5</sup>. Así, esta silogística intermedia añade las siguientes proposiciones de sentido común: *p* es la predominante afirmativa (*Pocos S no son P*), *b* es la predominante negativa (*Pocos S son P*), *t* es la mayoritaria afirmativa (*La mayoría de S es P*), *d* es la mayoritaria negativa (*La mayoría de S no es P*), *k* es la común afirmativa (*Muchos S son P*) y *g* es la común negativa (*Muchos S no son P*).

Ahora bien, como hemos visto, TFL provee un enfoque algebraico simple y correcto para el razonamiento silogístico que, desafortunadamente, no cubre casos de razonamiento de sentido común con cuantificadores no-clásicos como “mayoría”, “muchos” o “pocos”; por otro lado, la silogística extendida de Peterson y Thompson, SYLL<sup>+</sup>, incluye un rango más amplio de inferencias de sentido común en lenguaje natural, pero carece de un procedimiento algebraico. Así, dado este estado de cosas, en esta sección exponemos brevemente el sistema TFL<sup>+</sup> como una extensión de TFL que es capaz de lidiar con una amplia gama de inferencias de sentido común en lenguaje natural pero con las ventajas de un enfoque algebraico. Para exponer este sistema procedemos en tres pasos, primero proponemos una modificación de la sintaxis de TFL con el objetivo de representar los cuantificadores adicionales de SYLL<sup>+</sup>, posteriormente mostramos el método de decisión de TFL<sup>+</sup> y por último mencionamos en qué sentido este sistema es confiable.

Pues bien, primero, para representar las proposiciones *p*, *t*, *k*, *b*, *d* y *g* dentro del marco algebraico de TFL, consideremos la propuesta desplegada en el Cuadro 6.

**Tabla 6.** Representación de las proposiciones básicas.

SYLL <sup>+</sup>	TFL <sup>+</sup>	SYLL <sup>+</sup>	TFL <sup>+</sup>
SaP	$:= -S^0 + P^0$	SeP	$:= -S^0 - P^0$
SpP	$:= +S^3 + P^0$	SbP	$:= +S^3 - P^0$
StP	$:= +S^2 + P^0$	SdP	$:= +S^2 - P^0$
SkP	$:= +S^1 + P^0$	SgP	$:= +S^1 - P^0$
SiP	$:= +S^0 + P^0$	SoP	$:= +S^0 - P^0$

La razón detrás de esta propuesta es simple: de acuerdo con el marco lógico de SYLL<sup>+</sup>, las proposiciones intermedias no-universales, i.e., *p* (*b*), *t* (*d*) y *k* (*g*), son particulares hasta cierto punto, tal como lo son las proposiciones tipo *i* (*o*), lo cual nos obliga a elegir, siguiendo la sintaxis de TFL, una combinación  $+/-$  de términos para las proposiciones afirmativas; y una combinación  $+/-$  para las negativas. Sin embargo, esto no es suficiente porque, de acuerdo con SYLL<sup>+</sup>, las proposiciones *p* (*b*), *t* (*d*) y *k* (*g*) no son convertibles,<sup>6</sup> y por tanto, no son

<sup>5</sup> Aquí seguimos la presentación de [32].

<sup>6</sup> Así, por ejemplo, *t*  $:=$  *La mayoría de mexicanos hablan español* es particular, tal y como *i*  $:=$  *Algunos mexicanos hablan español*, pero claramente *t* no es convertible y

equivalentes a proposiciones de tipo *i* (*o*), lo cual nos obliga a usar algún tipo de bandera para denotar explícitamente este hecho: nosotros proponemos el uso de superíndices.

Ahora, de acuerdo con SYLL<sup>+</sup>, los nuevos cuantificadores implican un cierto orden (*p* (*b*) implica *t* (*d*), *t* (*d*) implica *k* (*g*) y *k* (*g*) implica *i* (*o*)) y por ende los superíndices se usan no sólo como banderas, sino como niveles ordenados de cuantificación. Esta elección sintáctica tiene las siguientes características: las proposiciones tipo *a*, *e*, *i* y *o* tienen nivel 0 para denotar el hecho de que se comportan de manera usual, como si no se hubieran hecho modificaciones; los superíndices se añaden a cada término con la finalidad de especificar el detalle de que las proposiciones tipo *p*, *t*, *k*, *b*, *d* y *g* no son convertibles; y además, estos índices nos permiten inducir un orden ( $3 \geq 2 \geq 1 \geq 0$ ) que indica que *a* (*e*) no entraña *p* (*b*), *t* (*d*), *k* (*g*), *i* (*o*); pero *p* (*b*), *t* (*d*), *k* (*g*) sí entrañan *i* (*o*).<sup>7</sup>

Ahora, dada esta representación, la modificación del método de decisión es como sigue: una conclusión se sigue válidamente de un conjunto de premisas *syss* *i*) la suma de las premisas es algebraicamente igual a la conclusión, *ii*) el número de conclusiones con cantidad particular (*viz.*, cero o uno) es igual al número de premisas con cantidad particular; y *iii*) el nivel de cuantificación de la conclusión es menor o igual que el máximo nivel de cuantificación de las premisas.

Para ejemplificar este procedimiento consideremos un par de ejemplos, uno válido (Tabla 7), uno inválido (Tabla 8: denotamos el hecho de que una conclusión no se sigue mediante “ $\not\vdash$ ”). Como es de esperarse, la adición de *p*, *t*, *k*, *b*, *d* y *g* incrementa el número de patrones o modos silogísticos correctos (*vide* Apéndice B).

Tabla 7. att-1

Proposición	TFL <sup>+</sup>
1. Todo H es M.	$-H^0 + M^0$
2. La mayoría de G son H.	$+G^2 + H^0$
$\vdash$ La mayoría de G son M.	$+G^2 + M^0$

Tabla 8. tta-1

Proposición	TFL <sup>+</sup>
1. La mayoría de H son M.	$+H^2 + M^0$
2. La mayoría de G son M.	$+G^2 + M^0$
$\not\vdash$ Todo G es M.	$-G^0 + M^0$

Como podemos ver, el sistema TFL<sup>+</sup> tiene las ventajas de un sistema algebraico (la reducción de un conjunto complejo de inferencias a un lenguaje formal

por tanto no es equivalente a *i*: notemos que si *Algunos mexicanos hablan español* entonces seguramente *Algunos hispanohablantes son mexicanos*, pero *La mayoría de mexicanos hablan español* no entraña que *La mayoría de hispanohablantes son mexicanos*. Contraejemplos similares pueden ser expuestos para mostrar que las proposiciones *p* (*b*), *t* (*d*) y *k* (*g*) no colapsan en proposiciones tipo *i* (*o*).

<sup>7</sup> Esto es diferente de la versión original de [32]: Thompson permite que las proposiciones universales entrañen proposiciones particulares, pero nuestra versión sigue la propuesta de Sommers y Englebreetsen, por lo que nosotros tenemos que añadir otra regla al marco SYLL<sup>+</sup>: si dos premisas son universales, la conclusión no puede ser particular (*vide* Apéndice B)

simple y unificado) y, al mismo tiempo, tiene las ventajas de una teoría inferencial con cuantificadores no-clásicos (la inclusión de un modelo de inferencia de sentido común en lenguaje natural), con la adición de que es confiable en el sentido de que el proceso inferencial es correcto:

**Proposición 1 (Confiabilidad)** *Una inferencia es  $\text{SYLL}_{\text{válida}}^+$  *sys* es  $\text{TFL}_{\text{válida}}^+$ .*

La relevancia de este sistema puede apreciarse mejor al considerar el compromiso entre las limitaciones expresivas de  $\text{TFL}$  y las limitaciones algebraicas de  $\text{SYLL}^+$  frente a la confiabilidad de  $\text{TFL}^+$  con respecto a inferencias de sentido común en lenguaje natural. A modo de ilustración, consideremos algunos ejemplos (Tabla 9- 12).

**Tabla 9.** kaa-1.

Proposición	$\text{TFL}^+$
1. Muchos H son I.	$+H^1 + I^0$
2. Este g es H.	$-g^0 + H^0$
$\not\vdash$ Este g es I.	$-g^0 + I^0$

**Tabla 10.** akt-4.

Proposición	$\text{TFL}^+$
1. Todo C es F.	$-C^0 + F^0$
2. Muchos M son C.	$+M^1 + C^0$
$\not\vdash$ La mayoría de M son F.	$+M^2 + F^0$

**Tabla 11.** bao-3.

Proposición	$\text{TFL}^+$
1. Pocos M son B.	$+M^3 - B^0$
2. Todo M es R.	$-M^0 + R^0$
$\vdash$ Algunos R no son B.	$+R^0 - B^0$

**Tabla 12.** etg-2.

Proposición	$\text{TFL}^+$
1. Ningún F es C.	$-F^0 - C^0$
2. La mayoría de V es C.	$+V^2 + C^0$
$\vdash$ Muchos V no son F.	$+V^1 - F^0$

#### 4.2. Una introducción a $\text{TFL}^{\text{PL}}$

Pues bien, si la relevancia de la confiabilidad de  $\text{TFL}^+$  tiene que ver con las limitaciones expresivas y algebraicas de  $\text{TFL}$  y  $\text{SYLL}^+$ —sin mencionar las limitaciones de  $\text{LPO}$ —con respecto a inferencias de sentido común en lenguaje natural, la utilidad de *Term Functor Logic Programming Language* ( $\text{TFL}^{\text{PL}}$ ) resulta de su potencial aplicación a sistemas de recuperación de información en los que la interacción en lenguaje natural con humanos es fundamental. Por estas consideraciones,  $\text{TFL}^{\text{PL}}$  es un lenguaje que, como Prolog, tiene una gramática especial. El siguiente fragmento es un ejemplo de programa en  $\text{TFL}^{\text{PL}}$ :

```
-s0+H0
-H0+M0
```



La primera línea puede leerse como “Sócrates es hombre” (i.e.  $\neg s^0 + M^0$  en TFL<sup>+</sup>) mientras que la segunda línea representa “Todo hombre es mortal” (i.e.  $\neg H^0 + M^0$  en TFL<sup>+</sup>). Esto permite ver que, así como en TFL la distinción entre proposición singular y universal desaparece, en TFL<sup>PL</sup> desaparece la distinción entre hechos y reglas que es usual, por ejemplo, en Prolog; esto también es diferente de la noción original de base de datos aristotélica de Mozes.

Ahora, por ejemplo, dado este programa, podemos llevar a cabo la siguiente consulta inferencial:

> s

```
-H0+M0
-s0+H0
-----
-s0+M0
```

esto es, “¿Qué es sócrates?” (s), y el programa responde  $\neg s0 + M0$ , es decir, “Sócrates es mortal”. A partir de este pequeño ejemplo podemos notar que, a diferencia de Prolog, la sintaxis de TFL<sup>PL</sup> es ternaria, aristotélica, y en ese sentido, TFL<sup>PL</sup> induce una base de datos *à la* Mozes; sin embargo, a diferencia de una base de datos aristotélica original, TFL<sup>PL</sup> rehuye la elección de una sintaxis binaria *à la* Frege (cf. §3) para favorecer una sintaxis más cercana al proyecto de Sommers. Todo esto resulta en la generación de un lenguaje de programación lógica basado en un sistema de lógica de términos y no en un sistema de primer orden, lo cual nos acerca a un lenguaje de programación lógica basado en una proyecto cognitivo de lógica natural.

#### 4.3. Sintaxis y semántica de TFL<sup>PL</sup>

Como resultado de estar basado en TFL<sup>+</sup>, la sintaxis de TFL<sup>PL</sup> es la misma de TFL<sup>+</sup> con el evidente cambio en la notación: el superíndice se escribe inmediatamente después de cada término. La sintaxis, entonces, se puede resumir mediante la siguiente BNF:

- $\langle \text{programa} \rangle ::= \langle \text{proposición} \rangle | \langle \text{proposición} \rangle \langle \text{programa} \rangle$
- $\langle \text{proposición} \rangle ::= \langle \text{término}^a \rangle \langle \pm T0 \rangle$
- $\langle \text{término}^a \rangle ::= \langle \pm T0 \rangle | \langle +T1 \rangle | \langle +T2 \rangle | \langle +T3 \rangle$

Un programa en TFL<sup>PL</sup> consiste de una o más proposiciones y cada proposición se define mediante dos términos definidos como en TFL<sup>+</sup>. Esta sintaxis formal nos permite definir, de modo riguroso y sin ambigüedad, los constructos de nuestro lenguaje, además de que permite apreciar la conexión directa entre la motivación filosófica y la motivación cognitiva.

Claramente, dada esta sintaxis, la semántica formal de TFL<sup>PL</sup> es directamente la semántica formal de TFL<sup>+</sup>, y dada la Proposición 1, es posible inferir que TFL<sup>PL</sup> preserva los resultados técnicos que pretendemos capturar, a saber, el proceso inferencial usando una lógica de términos.

#### 4.4. Implementación

La implementación de  $TFL^{PL}$  se ha realizado en C. Al día de hoy, el motor inferencial recibe un programa  $TFL^{PL}$  y hace un uso recursivo del procedimiento inferencial definido en la sección 4.1, a saber, que una conclusión se sigue válidamente de un conjunto de premisas syss *i)* la suma de las premisas es algebraicamente igual a la conclusión, *ii)* el número de conclusiones con cantidad particular (viz., cero o uno) es igual al número de premisas con cantidad particular; y *iii)* el nivel de cuantificación de la conclusión es menor o igual que el máximo nivel de cuantificación de las premisas.

#### 4.5. Ejemplo

```
-s0+M0 // Sócrates es hombre.
-f0+D0 // Fido es un perro.
-M0+A0 // Todo hombre es animal.
-D0+A0 // Todo perro es animal.
-M0+O0 // Todo hombre es mortal.
-A0+O0 // Todo animal es mortal.
+A3+O0 // Pocos animales no son mortales.

> s // Qué es Sócrates.
-M0+A0 // Todo hombre es animal.
-s0+M0 // Sócrates es hombre.
-----
-s0+A0 // Luego, Sócrates es animal.

-M0+O0 // Todo hombre es mortal.
-s0+M0 // Sócrates es hombre.
-----
-s0+O0 // Luego, Sócrates es mortal.
```

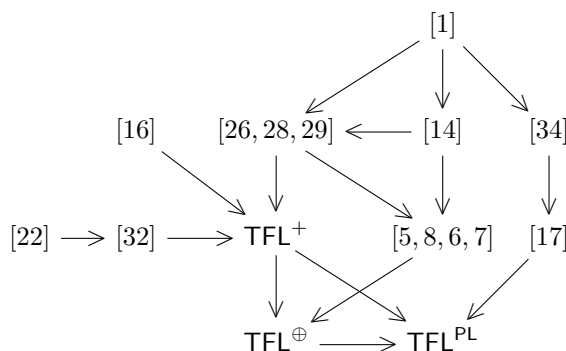
Algunas consideraciones que podemos extraer de la sintaxis, la implementación y el ejemplo anterior son las siguientes: *a)* como la sintaxis de  $TFL$  conlleva el abandono de la sintaxis Fregeana,  $TFL^{PL}$  no requiere del uso de variables y constantes individuales. *b)* Como la sintaxis de  $TFL$  conlleva la eliminación de la distinción entre proposición singular y proposición universal, la distinción entre hecho y regla desaparece en  $TFL^{PL}$ . *c)* Dadas las consideraciones anteriores,  $TFL$  no necesita hacer uso del predicado de igualdad, =, por lo que en  $TFL^{PL}$  no necesitamos hacer uso de algoritmos de sustitución y unificación como en Prolog. *d)*  $TFL^{PL}$  permite definir bases de datos aristotélicas en tanto que usa una base inferencial silogística y tiene una motivación más cercana a una lógica natural, si bien, a diferencia de una base de datos aristotélica original, su sintaxis no es binaria. *e)* Hasta el momento,  $TFL^{PL}$  sólo hace uso de proposiciones construidas mediante pares de términos, pero como  $TFL$  modela relaciones más complejas, es necesario añadir a  $TFL^{PL}$  un módulo relacional, un módulo para lidiar con

razonamiento numérico y, por supuesto, un módulo que regrese las respuestas en lenguaje natural.

## 5. Conclusiones

En esta contribución hemos alcanzado dos metas: *i)* la introducción de los elementos básicos del sistema lógico de términos TFL<sup>+</sup> y *ii)* la presentación de los avances de un lenguaje de programación diseñado a partir de tal sistema, TFL<sup>PL</sup>. [25] ha sugerido, por un lado, que la creación de un lenguaje de programación puede ocurrir por varias razones, como cuando una nueva área de aplicación demanda un nuevo lenguaje, o cuando las deficiencias de un lenguaje son notables, o cuando es más sencillo crear un nuevo lenguaje en lugar de modificar uno existente; y por otro lado, ha propuesto que un lenguaje es significativo cuando tal lenguaje es práctico y técnicamente novedoso. Pues bien, TFL<sup>PL</sup> surgió porque nos parece que, si bien hay lenguajes de programación lógica capaces de procesar lenguaje natural y realizar inferencia, no existe un lenguaje de programación basado en un proyecto de lógica natural, como se puede ver por la adopción de LPO por parte de los lenguajes de programación lógica (Cf. §2). Además, si bien hasta este momento TFL<sup>PL</sup> no parece más práctico que otros lenguajes de programación, ciertamente es técnicamente novedoso, tanto por su sintaxis como por su motivación. Y por ello, si bien TFL<sup>PL</sup> está en desarrollo, creemos que tiene potencial para el manejo de inferencias en lenguaje natural, lo cual es relevante dentro del paradigma clásico o fundacional de la inteligencia artificial: comprender y construir sistemas inteligentes.

Por último, nos gustaría cerrar este trabajo comentando que TFL<sup>PL</sup> forma parte de un proyecto cognitivo más general que incluye aspectos lógicos, lingüísticos, didácticos y filosóficos, y que se puede apreciar con mayor claridad de manera gráfica:



Consideramos, pues, que TFL<sup>PL</sup> es un sistema que promete no sólo como una herramienta computacional, sino como un dispositivo de investigación asociado a un proyecto cognitivo general en el que se podrán incluir módulos de lógica relacional (dado que TFL<sup>+</sup> es capaz de lidiar con inferencias relacionales),

razonamiento probabilístico (en tanto que  $TFL^+$  puede usarse para representar medidas probabilísticas (Cf. [33])) y razonamiento numérico (en la medida en que sea posible añadir módulos de inferencia silogística numérica (Cf. [18])); además de que podría adaptarse a estudios psicológicos (en la medida en que podría enriquecer las explicaciones psicológicas del razonamiento de sentido común (Cf. [11])) y filosóficos (en tanto sea capaz de promover la revisión de las lógicas de términos (Cf. [34,28,6,7]) como herramientas que podrían ser más interesantes y útiles de lo que originalmente habíamos creído (Cf.[4,10])).

## Apéndice A. Aspectos generales de la silogística

La *silogística* (SYLL) es una lógica de términos que tiene sus orígenes en *Primeros analíticos* [1] y estudia la relación de inferencia entre proposiciones categóricas. Una *proposición categórica* es una proposición compuesta por dos términos, una cantidad y una cualidad. El sujeto y el predicado de la proposición se llaman *términos*: el término-esquema S denota el término sujeto de la proposición y el término-esquema P denota el predicado. La *cantidad* puede ser universal (*Todo*) o particular (*Algún*) y la *cualidad* puede ser afirmativa (*es*) o negativa (*no es*).

Estas proposiciones categóricas se denotan mediante una *etiqueta* (a (para la universal afirmativa, SaP), e (para la universal negativa, SeP), i (para la particular afirmativa, SiP), y o (para la particular negativa, SoP)) que nos permite determinar una secuencia de tres proposiciones que se conoce como *modo*. Un *silogismo categórico*, entonces, es un modo ordenado de tal manera que dos proposiciones funcionan como premisas y la última como conclusión. Al interior de las premisas existe un término que ocurre en ambas premisas pero no en la conclusión: este término especial, usualmente denotado con el término-esquema M, funciona como un enlace entre los términos restantes y es conocido como término medio. De acuerdo a la posición del término medio se pueden definir cuatro arreglos o *figuras* que codifican los modos o patrones silogísticos válidos (Tabla 13)<sup>8</sup>.

**Tabla 13.** Modos silogísticos válidos

Figura 1	Figura 2	Figura 3	Figura 4
aaa	eae	iai	aee
eae	aee	aII	iai
aII	eio	oao	eio
eio	aoo	eio	

<sup>8</sup> Por mor de brevedad, pero sin pérdida de generalidad, omitimos los silogismos que requieren carga existencial.

## Apéndice B. Modos de la silogística intermedia

**Tabla 14.** Extensión de los modos silogísticos válidos (adaptado de [32])

	Figura 1	Figura 2	Figura 3	Figura 4
Con “mayoría”	aat	aed	ati	aed
	att	add	eto	eto
	ati	ado	tai	tai
	ead	ead	dao	
	etd	etd		
	eto	eto		
Con “muchos”	aak	aeg	aki	aeg
	atk	adg	eko	eko
	aki	ago	kai	kai
	akk	agg	gao	
	eag	eag		
	etg	etg		
	eko	eko		
	ekg	ehg		
Con “pocos”	aap	aeb	pai	aeb
	app	abb	epo	pai
	apt	abd	bao	epo
	apk	abg	api	
	api	abo		
	eab	eab		
	epb	epb		
	epd	epd		
	epg	epg		
	epo	epo		

Para los modos que necesitan carga existencial, como **aat-1** o **aak-1**, el único requisito que se necesita para producir una inferencia válida en  $TFL^+$  es añadir la premisa implícita que enuncia la existencia del término sujeto, es decir, algo como  $+S + S$ .

## Referencias

1. Aristotle, Smith, R.: Prior Analytics. Hackett Classics Series, Hackett (1989)
2. Bergin, T., Gibson, R.: History of Programming Languages II. ACM Press books, ACM Press (1996)
3. Bratko, I.: Prolog Programming for Artificial Intelligence. International computer science series, Addison Wesley (2001)
4. Carnap, R.: Die alte und die neue logik. Erkenntnis 1, 12–26 (1930), <http://www.jstor.org/stable/20011586>

5. Englebretsen, G.: The New Syllogistic. 05, P. Lang (1987)
6. Englebretsen, G.: Something to Reckon with: The Logic of Terms. Canadian electronic library: Books collection, University of Ottawa Press (1996)
7. Englebretsen, G., Sayward, C.: Philosophical Logic: An Introduction to Advanced Topics. Bloomsbury Academic (2011)
8. Englebretsen, G.: Linear diagrams for syllogisms (with relationals). *Notre Dame J. Formal Logic* 33(1), 37–69 (12 1991), <https://doi.org/10.1305/ndjfl/1093636009>
9. Geach, P.: Reference and Generality: An Examination of Some Medieval and Modern Theories. *Contemporary Philosophy / Cornell University*, Cornell University Press (1962)
10. Geach, P.: Logic Matters. Campus (University of California Press), University of California Press (1980)
11. Khemlani, S., Johnson-laird, P.N.: Theories of the syllogism: a meta-analysis. *Psychological Bulletin* pp. 427–457 (2012)
12. Kowalski, R.A.: The early years of logic programming. *Commun. ACM* 31(1), 38–43 (Jan 1988), <http://doi.acm.org/10.1145/35043.35046>
13. Kuhn, S.T.: An axiomatization of predicate functor logic. *Notre Dame J. Formal Logic* 24(2), 233–241 (04 1983), <https://doi.org/10.1305/ndjfl/1093870313>
14. von Leibniz, G., Couturat, L.: Opuscles et fragments inédits de Leibniz: extraits des manuscrits de la Bibliothèque royale de Hanovre. Olms paperback, Olms (1961)
15. Moss, L.: Natural logic. In: Lappin, S., Fox, C. (eds.) *The Handbook of Contemporary Semantic Theory*. John Wiley & Sons (2015)
16. Mostowski, A.: On a generalization of quantifiers. *Fundamenta Mathematicae* 44(2), 12–36 (1957)
17. Mozes, E.: A deductive database based on aristotelian logic. *Journal of Symbolic Computation* 7(5), 487–507 (1989), <http://www.sciencedirect.com/science/article/pii/S0747717189800306>
18. Murphree, W.A.: Numerical term logic. *Notre Dame J. Formal Logic* 39(3), 346–362 (07 1998), <https://doi.org/10.1305/ndjfl/1039182251>
19. Nakatsu, R.T.: Diagrammatic Reasoning in AI. Wiley (2009)
20. Noah, A.: Predicate-functors and the limits of decidability in logic. *Notre Dame J. Formal Logic* 21(4), 701–707 (10 1980), <https://doi.org/10.1305/ndjfl/1093883255>
21. Noah, A.: Sommers’s cancellation technique and the method of resolution. In: Oderberg, D. (ed.) *The Old New Logic: Essays on the Philosophy of Fred Sommers*, pp. 169–182. a Bradford book (2005)
22. Peterson, P.L.: On the logic of “few”, “many”, and “most”. *Notre Dame J. Formal Logic* 20(1), 155–179 (01 1979), <https://doi.org/10.1305/ndjfl/1093882414>
23. Quine, W.V.O.: Predicate functor logic. In: Fenstad, J.E. (ed.) *Proceedings of the Second Scandinavian Logic Symposium*. North-Holland
24. Russell, B.: A critical exposition of the philosophy of Leibniz: with an appendix of leading passages. G. Allen & Unwin (1937)
25. Sammet, J.E.: Programming languages: History and future. *Commun. ACM* 15(7), 601–610 (Jul 1972), <http://doi.acm.org/10.1145/361454.361485>
26. Sommers, F.: On a fregean dogma. In: Lakatos, I. (ed.) *Problems in the Philosophy of Mathematics, Studies in Logic and the Foundations of Mathematics*, vol. 47, pp. 47–81. Elsevier (1967), <http://www.sciencedirect.com/science/article/pii/S0049237X08715210>
27. Sommers, F.: Intellectual autobiography. In: Oderberg, D. (ed.) *The Old New Logic: Essays on the Philosophy of Fred Sommers*, pp. 1–24. A Bradford book (2005)

28. Sommers, F.: The Logic of Natural Language. Clarendon Library of Logic and Philosophy, Clarendon Press; Oxford: New York: Oxford University Press (1982)
29. Sommers, F., Englebretsen, G.: An Invitation to Formal Reasoning: The Logic of Terms. Ashgate (2000)
30. Staal, J.F.: Formal logic and natural languages (a symposium). Foundations of Language 5(2), 256–284 (1969), <http://www.jstor.org/stable/25000379>
31. Sterling, L., Shapiro, E.: The Art of Prolog: Advanced Programming Techniques. Logic programming, MIT Press (1994)
32. Thompson, B.: Syllogisms using “few”, “many”, and “most”. Notre Dame J. Formal Logic 23(1), 75–84 (01 1982), <https://doi.org/10.1305/ndjfl/1093883568>
33. Thompson, B.: Syllogisms with statistical quantifiers. Notre Dame J. Formal Logic 27(1), 93–103 (01 1986), <https://doi.org/10.1305/ndjfl/1093636527>
34. Veatch, H.B.: Intentional logic: a logic based on philosophical realism. Archon Books (1970)
35. Woods, J.: Logic Naturalized, pp. 403–432. Springer International Publishing, Cham (2016), [https://doi.org/10.1007/978-3-319-26506-3\\_18](https://doi.org/10.1007/978-3-319-26506-3_18)





## Aplicación de lógica difusa en el proceso de *shot peening* del aluminio 2024-T351

Alicia Guadalupe Lazcano Herrera<sup>1</sup>, Sandra Silvia. Roblero Aguilar<sup>1,2</sup>,  
José Solís Romero<sup>1</sup>, Héctor Rafael Orozco Aguirre<sup>2</sup>,  
Víctor Augusto Castellanos Escamilla<sup>1</sup>

<sup>1</sup> SEP/SES/TecNM/Instituto Tecnológico de Tlalnepantla, Estado de México,  
México

<sup>2</sup> Centro Universitario UAEM Valle de México, Estado de México,  
México

{alicia\_gpe\_hg, ssrauaemex}@hotmail.com, {jsolis,vcastellanos}@ittla.edu.mx,  
rafilla.orozco@gmail.com

**Resumen.** El proceso de granallado o shot peening se emplea rutinariamente para incrementar la resistencia a la fatiga de componentes y estructuras. Los efectos que se producen en la superficie del material tratado son esfuerzos compresivos residuales, rugosidad superficial y endurecimiento por deformación, los cuales dependen de la correcta elección de los parámetros de procesamiento. En esta investigación se aplica el método de lógica difusa para determinar la mejor combinación de factores de control que inciden en las respuestas multi-objetivo de una aleación de aluminio 2024-T351 (AA) tratada con shot peening. Los factores de control de entrada son el tipo de bolilla (shot), la cobertura y el ángulo de incidencia. Para fines estadísticos, los parámetros experimentales se trabajaron utilizando un arreglo fraccional ortogonal  $L_{16}$ . Las tres propiedades que se determinaron en forma experimental directamente sobre AA tratada son las entradas para el sistema de inferencia difuso y la salida es el índice de respuesta (IR). Se realizó una comparación del IR entre las condiciones de tratamiento inicial con las óptimas, resultando una mejora en la resistencia a la fatiga.

**Palabras clave:** Lógica difusa, sistema de inferencia difuso, shot peening/Granallado

## Application of Fuzzy Logic in the Process of Shot Peening of Aluminium 2024-T351

**Abstract.** Shot peening is widely used to enhance the fatigue properties of components and structures. Compressive residual stresses, surface roughness,

and work hardening are the immediate effects induced on the surface/subsurface layer of the treated material, which depend on the correct choice of the peening process parameters. This research aims to apply the fuzzy logic method for determining the best optimal selection of the control factors that directly influence the multi-objective response properties of a peened 2024-T351 aluminium alloy (AA). The input parameters taken into consideration are shot, coverage and incidence angle. For statistical purposes, the experimental parameters were put in place with a  $L_{16}$  orthogonal fractional array. The three induced properties extracted experimentally from the treated AA are fed as inputs to fuzzy inference system and output withdrew is the response index (IR). A comparison of the IR between the initial and optimal peening conditions shows an improvement in fatigue resistance.

**Keywords:** Fuzzy logic, fuzzy inference system, shot peening.

## 1. Introducción

Los componentes utilizados tanto en la industria aeroespacial como en la industria automotriz a menudo están sujetos a condiciones de carga fluctuantes o cíclicas, dando origen a la fatiga de componentes. Las grietas por fatiga se originan normalmente desde la superficie de la parte debido, entre otros, a defectos contenidos en el material [1]. Cuando la carga dinámica continúa, la grieta crece hasta que alcanza una longitud tal que ninguna barrera le podrá disminuir su velocidad de crecimiento o detener. Sin embargo, cuando la grieta creciente se obstruye en su camino por diversas barreras muy compactas, como ocurriría en microestructuras que contienen granos pequeños, entonces el crecimiento de la grieta disminuye la velocidad de crecimiento debido a que su trayectoria se desvía lo que resulta en un incremento en la resistencia a la fatiga. Es por ello que la resistencia a la fatiga usualmente se cuantifica como la resistencia a la propagación de grietas ofrecida por el material [2].

Para acondicionar la superficie en forma tal que permita detener o disminuir la velocidad de propagación de grietas, el proceso de granallado o *shot peening* (SP) es una alternativa sencilla y económica, en donde se introducen muchas y variadas formas de barreras de textura o distorsión microestructural en la superficie y capas sub-superficiales. En este proceso, esencialmente, bolillas esféricas conocidas como shots, que están hechas de acero al carbón, hierro, acero inoxidable, vidrio o trozos redondeados de cerámico, se proyectan en forma deliberada para golpear la superficie de un componente metálico. Las bolillas se aceleran ya sea por medio de aire comprimido o por fuerzas centrífugas. Las velocidades al impacto son suficientes para que se origine una indentación (huella) en la superficie del material tratado, en donde el régimen que domina es completamente plástico. Precisamente ese régimen plástico da origen a un estado de esfuerzos residuales compresivos y un endurecimiento por deformación en la sub-superficie, los cuales forman parte de esas barreras que inherentemente obstruyen el libre movimiento de las grietas.

En el proceso de SP la magnitud de los parámetros de procesamiento, así como todos los efectos inducidos deben controlarse a fin de lograr un beneficio en términos de resistencia a la fatiga y evitar o reducir la introducción de daño severo (por ejemplo,

iniciación de grieta causada por una excesiva rugosidad) [3, 4]. Por lo tanto, el desarrollo de un método rápido y eficiente para optimizar el proceso es una importante área de estudio para realizar investigación. Bajo este contexto, el diseño de experimentos estadístico (DoE por sus siglas en inglés) está considerado como una herramienta consolidada para resolver problemas relacionados con la elección de los parámetros óptimos de procesamiento [5]. Unal [6] utilizó el DoE para la optimización de los parámetros en SP sobre probetas estandarizadas llamadas tiras Almen.

La optimización se realizó extrayendo los niveles óptimos de cada propiedad y posteriormente se compilaron los parámetros de procesamiento. Por otro lado, George [7] aplicó la metodología DoE con Taguchi para lograr la optimización de parámetros críticos, así como establecer el orden de predominancia, utilizando el análisis de varianza (ANOVA) para finalmente predecir un arreglo óptimo de cada parámetro. En breve, el DoE emplea diseños ortogonales para examinar las características de calidad por medio de un número reducido de experimentos. Sin embargo, esta técnica está limitada cuando se trata del tratamiento de problemas multi-respuesta, en otras palabras, el nivel óptimo de los parámetros de procesamiento es verdadero para la optimización de una respuesta individual del proceso.

Para contrarrestar lo anterior, la técnica de lógica difusa [8] tiene la capacidad de trabajar con diversas entradas, como por ejemplo, características funcionales de procesos, para eventualmente convertir el comportamiento múltiple en un solo índice multi-respuesta (IR) [9]. El procedimiento de cómputo suave como la lógica difusa está ganando una aceptación progresiva en una variedad de situaciones ingenieriles [10-12], en virtud de su utilidad cuando no se encuentra disponible información matemática exacta. En comparación con otros métodos de la inteligencia artificial, el desarrollo de la lógica difusa es moderadamente más sencillo y no necesita de mucho software ni demasiado hardware. Sin embargo, hasta donde los autores tienen conocimiento, no se reportan trabajos dedicados a la aplicación de esta metodología como una estrategia para optimizar el proceso de SP con la intención de incrementar la resistencia a la fatiga de materiales metálicos.

En el presente trabajo, se emplea un mecanismo de inferencia difuso para determinar los parámetros óptimos de procesamiento en el tratamiento superficial del aluminio 2024-T351, utilizando las respuestas experimentalmente determinadas como son los esfuerzos residuales, el endurecimiento por deformación y la rugosidad superficial en términos de concentración de esfuerzos.

## **2. Generalidades de la lógica difusa**

La lógica difusa (LD) está relacionada y fundamentada en la teoría de los conjuntos difusos, en la cual, el grado de pertenencia de un elemento a un conjunto está determinado por una función de pertenencia (FP) [13-15] que puede adoptar valores reales comprendidos en el intervalo  $[0,1]$ . De esta manera, mientras que en el marco de la lógica clásica un parámetro tiene pertinencia o no, dándole un valor de 1 si es pertinente y 0 en caso contrario; En la LD, se obtiene un nivel de cumplimiento de la



**Fig. 1.** Esquema general de funcionamiento de un sistema de inferencia difuso.

pertinencia, es decir, entre más cercano a cero, será menos pertinente y cuando sea más cercano a 1 será más pertinente.

Las fases que completan el montaje de un sistema de inferencia difuso Mandani [15, 16] se muestran en la Fig. 1. Textualmente se describen como sigue:

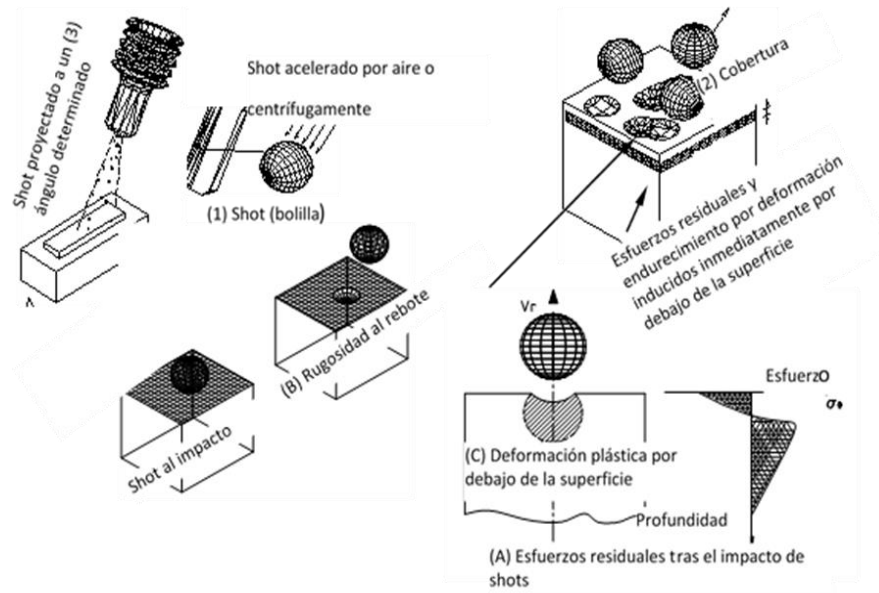
- Fase 1: se basa en un proceso donde las variables tienen un grado de incertidumbre metalingüístico. Es decir, el rango de valores de cada variable puede clasificarse por conjuntos difusos, originando el universo del discurso. Con ello, los valores pasan a un proceso de fusificación que los categoriza en un nivel de pertenencia entre 0 y 1 que pertenece a un conjunto difuso. Los conjuntos difusos son caracterizados mediante funciones de pertenencia, las cuales están sintonizadas al punto de operación adecuado para el funcionamiento del sistema, es decir, las reglas de inferencia que serán empleadas
  - Fase 2: se proponen reglas lingüísticas conocidas como de inferencia. Con esto, el grado de pertenencia de cada una de las variables se evalúa en un subconjunto de estas reglas.
- Fase 3: aquí se determinan los valores óptimos de salida, mediante un mecanismo conocido como defusificación, el cual consiste en pasar el grado de pertenencia, proveniente de la consecuencia de la regla de inferencia activada, a un valor nítido o real, con el fin de obtener un valor cuantificable.

### 3. Caso de estudio: aplicación de lógica difusa en la selección óptima de parámetros del proceso de *shot peening*

El desarrollo del presente estudio se llevó a cabo bajo la consideración de las siguientes etapas: recopilación de datos, fusificación, generación de reglas y la aplicación del sistema de inferencia difuso (FIS, por sus siglas en inglés).

#### 3.1. Recopilación de datos

En general, del proceso SP previamente descrito en la introducción, la información que se utiliza en el presente análisis se puede clasificar como (i) los factores de control de procesamiento y (ii) las propiedades de respuesta o efectos inducidos. Los factores de control que se eligieron son el tipo/tamaño de bolilla o shot (1), la cobertura en porcentaje (2) y el ángulo de incidencia o impacto (3). Por su parte, las propiedades de



**Fig. 2.** Representación esquemática para ilustrar los parámetros de proceso y los efectos inducidos por el shot peening.

respuesta experimentalmente determinados son los (A) esfuerzos residuales, (B) rugosidad en términos de concentraciones de esfuerzos y (C) endurecimiento por deformación. Con la intención de clarificar los datos que intervienen en el proceso, en la Fig. 2 se ilustra cada uno de los factores y efectos que modifican el estado superficial de un componente.

En los tres factores de control o de procesamiento que se consideraron, se evaluaron cuatro niveles de acción, como se muestra en la Tabla 1. Los factores elegidos son de fundamental importancia para la industria aeroespacial y automotriz [17].

Basándose en los diferentes factores y niveles, y utilizando la metodología diseño de experimentos, se eligió un arreglo ortogonal factorial fraccionado  $L_{16}(4)^5$ . Este enfoque permite alcanzar conclusiones que son válidas en un rango de condiciones experimentales, además de que representa una elección razonable en términos de la reducción del número de experimentos sin pérdida de calidad en la obtención de resultados. El arreglo ortogonal se ilustra en la Tabla 2.

Cabe mencionar que el arreglo ortogonal está construido para usarse con cinco factores, por lo que se necesitó dejar dos columnas vacías.

Esta acción apoyó en forma positiva el diseño experimental porque todos los parámetros que influyen los efectos del proceso son considerados.

Las propiedades de respuesta que resultan de los experimentos de acuerdo con la Tabla 2, se muestran en la Tabla 3. Los esfuerzos residuales se determinaron utilizando el método del agujero ciego, mientras que para el endurecimiento por deformación se

**Tabla 1.** Factores y sus respectivos niveles de control.

	Factor	Nivel del factor			
		1	2	3	4
A	Tipo de shot	S230	CCW20	S110	S330
B	Cobertura	50	100	200	400
D	Ángulo de incidencia	30	90	45	90

**Tabla 2.** Arreglo ortogonal para la experimentación.

Arreglo ortogonal						Factores asignados y sus niveles				
Exp.	A	B	C	D	E	SHOT (A)	Cobertura (%) (B)	Vacío (C)	Ángulo (°) (D)	Vacío (E)
1	1	1	1	1	1	S230	50	—	30	—
2	1	2	2	2	2	S230	100	—	90	—
3	1	3	3	3	3	S230	200	—	45	—
4	1	4	4	4	4	S230	400	—	90	—
5	2	1	2	3	4	CCW20	50	—	45	—
6	2	2	1	4	3	CCW20	100	—	90	—
7	2	3	4	1	2	CCW20	200	—	30	—
8	2	4	3	2	1	CCW20	400	—	90	—
9	3	1	3	4	2	S110	50	—	90	—
10	3	2	4	3	1	S110	100	—	45	—
11	3	3	1	2	4	S110	200	—	90	—
12	3	4	2	1	3	S110	400	—	30	—
13	4	1	4	2	3	S330	50	—	90	—
14	4	2	3	1	4	S330	100	—	30	—
15	4	3	2	4	1	S330	200	—	90	—
16	4	4	1	3	2	S330	400	—	45	—

usó un durómetro Vickers y para la concentración de esfuerzos se consideró un perfilómetro. Para los detalles específicos, se puede consultar [18].

**Tabla 3.** Propiedades de respuesta experimental.

No. Exp.	Esfuerzos residuales (Mega Pascales)	Endurecimiento (Dureza Vickers)	Concentración de esfuerzos (adimensional)
	ER	ED	CE
1	-124.80	144.61	1.36
2	-233.70	145.15	1.55
3	-197.20	154.74	1.66
4	-246.65	166.90	1.62
5	-173.15	135.65	1.56
6	-243.00	147.45	1.72
7	-164.95	150.08	1.63
8	-245.55	157.55	1.84
9	-280.00	145.26	1.55
10	-131.50	153.50	1.58
11	-379.20	157.50	1.57
12	-149.37	151.75	1.53
13	-257.90	134.50	1.45
14	-170.00	159.00	1.43
15	-308.50	157.89	1.57
16	-292.30	153.58	1.54

### 3.2. Fusificación

Establecidos los datos, se procede a definir las variables y valores lingüísticos, así como la función de pertenencia para realizar la fusificación. Dos subconjuntos difusos (bajo y alto) se asignaron uniformemente en tres variables de entrada, designadas como: Esfuerzo Residual (ER), Concentración de Esfuerzos (CE) y endurecimiento por deformación (ED). Bajo esta premisa, los valores que se asignaron en términos de pertenencia se compilan en la Tabla 4.

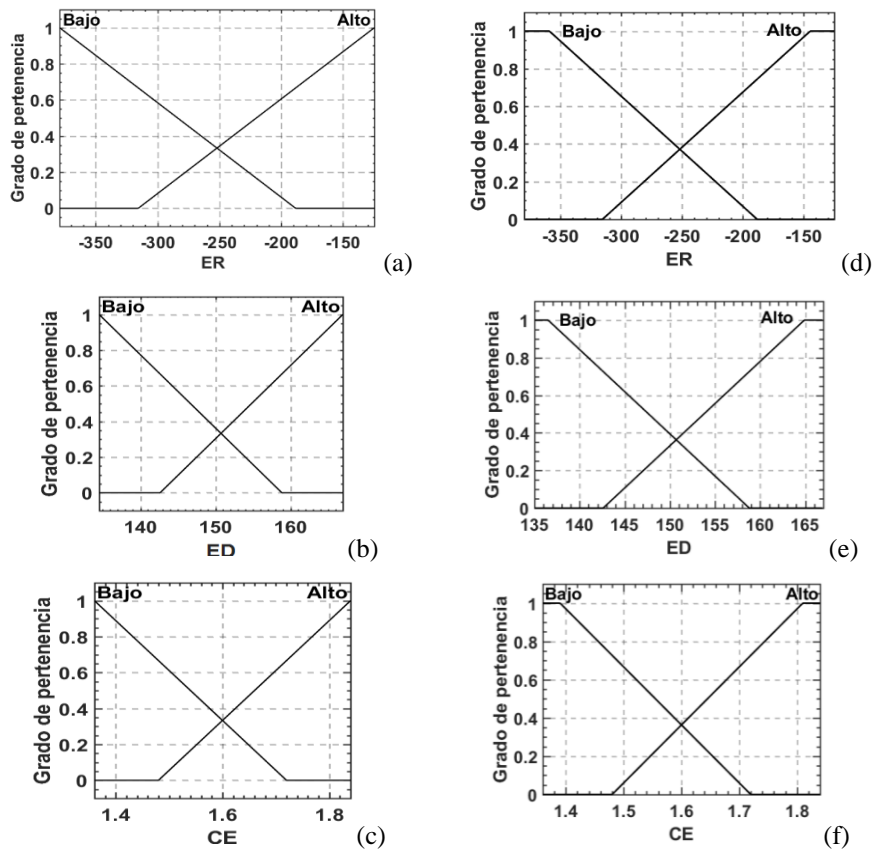
Los valores de los datos de entrada se deben definir como FPs. Una FP puede adoptar diferentes formas: trapezoidal, triangular sigmoïdal, gaussiana, o una combinación de formas para cada entrada. Las FPs triangulares y trapezoidales se consideraron para las entradas debido a su frecuencia de uso en aplicaciones de ingeniería, con la intención de encontrar los mejores resultados entre ellas. En la Fig. 3 se representa gráficamente las funciones de pertenencia para las tres entradas, las cuales se generaron por medio del software comercial Matlab.

### 3.3. Definición de las reglas difusas

El método Mandani se eligió como el motor de inferencia, el cual se basa en la colección de reglas de control del tipo *si-entonces*, y (*if-then*, *and*, por su nombre en inglés, respectivamente) las cuales se definieron con la siguiente configuración:

**Tabla 4.** Funciones de pertenencia de las propiedades respuesta con valores nítidos.

Variable Lingüística	Valores Lingüísticos	Función de pertenencia Triangular	Función de pertenencia Trapezoidal
<b>ER (<math>K_1</math>)</b>	Bajo ( $L_1$ )	$\mu_{l_1}^{k_1}: (-570, -379.2, -188.4)$	$\mu_{l_1}^{k_1}: (-570, -379.2, -359.2, -188.4)$
	Alto ( $L_2$ )	$\mu_{l_2}^{k_1}: (-315.6, -124.8, 315.6)$	$\mu_{l_2}^{k_1}: (-315.6, -144.8, -124.8, 315.6)$
<b>ED(<math>K_2</math>)</b>	Bajo ( $L_1$ )	$\mu_{l_1}^{k_2}: (110.2, 134.5, 158.8)$	$\mu_{l_1}^{k_2}: (110.2, 134.5, 136.5, 158.8)$
	Alto ( $L_2$ )	$\mu_{l_2}^{k_2}: (142.6, 166.9, 191.2)$	$\mu_{l_2}^{k_2}: (142.6, 164.9, 166.9, 191.2)$
<b>CE (<math>K_3</math>)</b>	Bajo ( $L_1$ )	$\mu_{l_1}^{k_3}: (1.0, 1.36, 1.72)$	$\mu_{l_1}^{k_3}: (1.0, 1.36, 1.39, 1.72)$
	Alto ( $L_2$ )	$\mu_{l_2}^{k_3}: (1.48, 1.84, 2.2)$	$\mu_{l_2}^{k_3}: (1.48, 1.81, 1.84, 2.2)$



**Fig. 3.** Funciones de pertenencia de las propiedades de respuesta como variables de entrada: triangulares (a-c), y trapezoidales (d-f). Triangular (a) y trapezoidal (b).

*Regla 1: if ER es Bajo and ED es Bajo and CE es Bajo then IR es MuyBajo*  
*Regla 2: if ER es Bajo and ED es Bajo and CE es Alto then IR es Bajo*



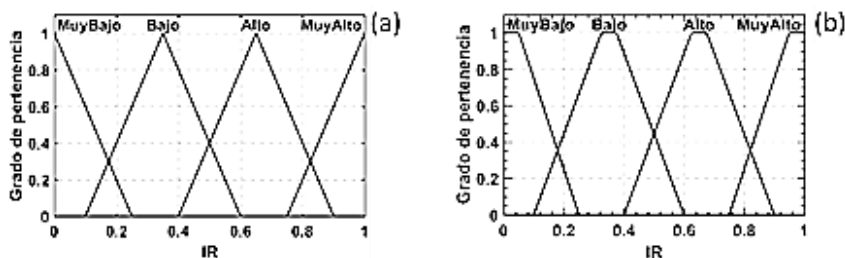


Fig. 4. Funciones de pertenencia de la salida en términos del índice de respuesta (IR).

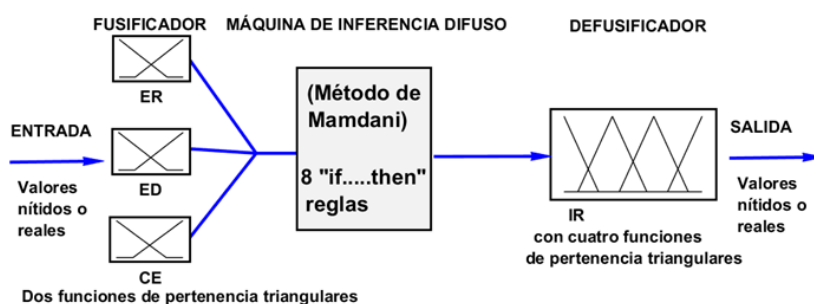


Fig. 5. Representación esquemática de la arquitectura de la unidad de inferencia difusa, ejemplificada con las funciones de pertenencia triangulares.

Regla 3: *if ER es Bajo and ED es Alto and CE es Bajo then IR es Bajo*  
 Regla 4: *if ER es Bajo and ED es Alto and CE es Alto then IR es Alto*  
 Regla 5: *if ER es Alto and ED es Bajo and CE es Bajo then IR es Bajo*  
 Regla 6: *if ER es Alto and ED es Bajo and CE es Alto then IR es Alto*  
 Regla 7: *if ER es Alto and ED es Alto and CE es Bajo then IR es Alto*  
 Regla 8: *if ER es Alto and ED es Alto and CE es Alto then IR es MuyAlto*

### 3.4. Aplicación del sistema de inferencia difuso

Para la defusificación se definió una variable de salida, designada como índice de respuesta (IR), con cuatro valores lingüísticos: Muy Bajo, Bajo, Alto y Muy Alto, como se puede apreciar en la Fig. 4.

Es importante señalar que para la defusificación se emplea el centro de gravedad (COG), lo cual implica que el valor a obtener para el índice IR se ubica en el centro del rango de pertenencia. En la Fig. 5 se muestra el esquema del modelo difuso propuesto descrito en este apartado, tomando como ejemplo un elemento en cada etapa.

## 4. Análisis de resultados

El índice IR determinado de las propiedades multi-respuesta se obtiene utilizando la herramienta de lógica difusa que trabaja con el software comercial Matlab. Los valores

**Tabla 5.** Salida de FIS mostrando los IRs como valores nítidos.

Exp. No.	Valores nítidos de entrada			Valores nítidos de salida (IRs)	
	ER	ED	CE	FP:Triangular	FP: Trapezoidal
1	-124.8	144.61	1.36	0.392	0.391
2	-233.7	145.15	1.55	0.424	0.422
3	-197.2	154.74	1.66	0.647	0.65
4	-246.65	166.9	1.62	0.592	0.594
5	-173.15	135.65	1.56	0.454	0.453
6	-243	147.45	1.72	0.561	0.56
7	-164.95	150.08	1.63	0.593	0.593
8	-245.55	157.55	1.84	0.696	0.698
9	-280	145.26	1.55	0.388	0.387
10	-131.5	153.5	1.58	0.599	0.599
11	-379.2	157.5	1.57	0.455	0.455
12	-149.37	151.75	1.53	0.546	0.546
13	-257.9	134.5	1.45	0.267	0.267
14	-170	159	1.43	0.65	0.65
15	-308.5	157.89	1.57	0.466	0.466
16	-292.3	153.58	1.54	0.423	0.424

numéricos nítidos se muestran en la Tabla 5. Evidentemente los valores numéricos de cada función de pertenencia utilizada prácticamente no presentan discrepancia significativa, lo que claramente indica que las formas de pertenencia lineales no influyen en el tratamiento de entradas multi-respuesta a fin de alcanzar un índice de respuesta.

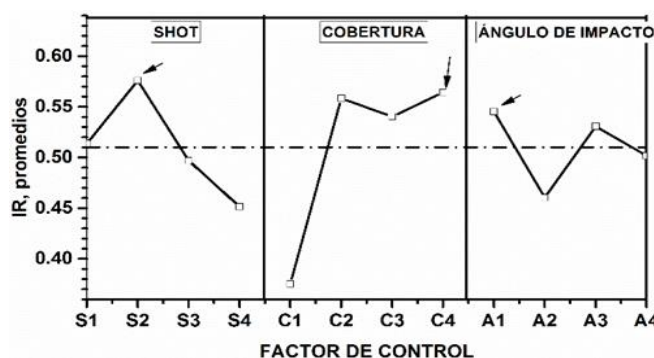
Entre más alto sea el valor del IR más alta la posibilidad de que esa corrida experimental represente la óptima. Al respecto, la corrida experimental No. 8 adopta las mejores características de rendimiento múltiple de entre los 16 experimentos para ambas funciones de pertenencia.

La ortogonalidad hace posible extraer el efecto de cada uno de los parámetros de los datos experimentales a sus diferentes niveles, porque cuando se determina el promedio para un nivel de factor, los otros factores en cada uno de sus niveles toman un número equivalente de veces. Así, el promedio de IR para cada uno de los niveles de los parámetros de prueba, al igual que el promedio total de los IRs se muestra en la Tabla 6.

En la tabla se exhiben valores delta (máx.-mín.), cuyo número más alto es indicativo del nivel de significancia de un factor en particular. Con la intención de visualizar el grado de significancia de cada parámetro sobre los efectos del SP, en la gráfica de la Fig. 6 se muestra el comportamiento de cada parámetro en términos de los promedios IR por nivel. En este caso, inmediatamente se puede deducir que la cobertura tiene la contribución más alta seguida por el shot y al final el ángulo de incidencia. De la misma tabla de respuesta es posible extraer los parámetros que podrían ofrecer la condición óptima para incrementar la resistencia a la fatiga. De acuerdo con la gráfica, la

**Tabla 6.** Valores promedio IRs para cada nivel de cada factor.

Factores de control			
Nivel	Shot	Cobertura	Ángulo de incidencia
1	0.51375	0.37525	0.54525
2	0.576	0.5585	0.4605
3	0.497	0.54025	0.53075
4	0.4515	0.56425	0.50175
Max-min	0.1245	0.189	0.08475
Promedio total de IR = 0.510			



**Fig. 6.** Factores de control promedio con sus respectivos niveles en términos del IR.

combinación óptima de parámetros, señalada por las flechas, corresponde a S2, C4 y A1, es decir, de la Tabla 1, shot (A2)=CW20, cobertura (B4) = 400 % y ángulo de incidencia (D1) = 30°. Como se esperaba, este resultado confirma el que se determinó en la Tabla 5, resaltando el experimento 8 como el óptimo, el cual tiene el mismo orden de factores y niveles.

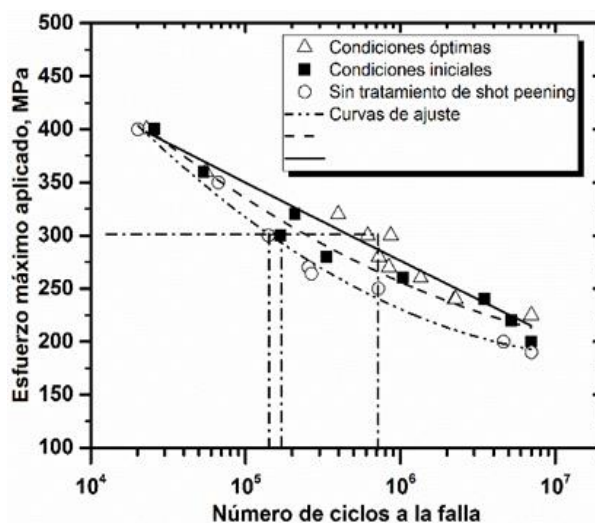
#### 4.1. Pruebas de confirmación

Toda vez que se determinó la combinación óptima de parámetros de procesamiento con base a los efectos que provoca el SP considerados aquí como la multi-respuesta o tres funciones objetivo, se procede a realizar la comparación con la corrida experimental inicial. Los resultados se presentan en la Tabla 7.

Los resultados experimentales exhiben una ligera ganancia en los esfuerzos residuales (ER), quedando prácticamente el mismo nivel de endurecimiento por deformación (ED), pero con un incremento en concentración de esfuerzos (CE) en las condiciones iniciales respecto a la combinación óptima. Este último resultado repercute negativamente en la resistencia a la fatiga, lo que significa que las condiciones óptimas efectivamente resultaron en una mejora significativa, como se puede apreciar en la Fig. 7. Puede observarse, por ejemplo, que, a 300 MPa de esfuerzo aplicado, la probeta sin tratamiento de SP se fractura (falla) por los 150000 ciclos, entre tanto, la probeta con

**Tabla 7.** Resultados experimentales bajo condiciones iniciales y óptimas.

Efectos	Combinación óptima	Resultados experimentales	Condiciones iniciales	Resultados experimentales
ER	A2, B4, D1 (CW20,400%, 30°)	300 MPa	A4, B4, D4 (S330,400%,90°)	373 MPa
ED		147 HV		150 HV
CE		1.3		1.8



**Fig. 7.** Curvas S-N, bajo carga axial y amplitud constante para probetas con SP y sin SP.

las condiciones iniciales (sin optimizar) fracturó por los 160000 ciclos y la probeta con la que se aplicó tratamiento empleando las condiciones determinadas con lógica difusa se fracturó por los 700000 ciclos. La resistencia a la falla por fatiga se incrementó hasta por un 300% con respecto a las condiciones iniciales.

## 5. Conclusiones y trabajo futuro

En relación con los resultados alcanzados en las pruebas de confirmación, se pueden configurar las siguientes conclusiones:

- El modelo de inferencia difuso es efectivo para tratar con aplicaciones ingenieriles multi-respuesta, en forma sencilla y con bajo costo. El proceso de shot peening, así como otros procesos de la ingeniería de superficies (como recubrimientos) son factibles para implementar el uso de la inteligencia artificial.
- En el presente estudio, el proceso de SP, se determinó una combinación de factores de control, mediante los cuales se generaron las pruebas de fatiga que

resultaron con un incremento apreciable en la resistencia a la fatiga del aluminio 2024-T351, comúnmente utilizado en estructuras de aeronaves.

- La metodología asociada entre el diseño de experimentos y lógica difusa ofrece una alternativa positiva en la reducción de tiempo y costo en las investigaciones que tratan con resultados no lineales ya que evita el típico trato de prueba y error, así como la realización de una gran cantidad de experimentos que no ofrecen mejores resultados.
- Las funciones de pertenencia lineales no presentan discrepancia significativa cuando se trabaja con respuestas multi-objetivo, por lo que resulta indistinto utilizar la función de pertenencia triangular o trapezoidal para este tipo de aplicaciones.

La aplicación de la técnica de cómputo suave permitió integrar tres respuestas con el resultado favorable de una sola salida, lo que posibilitó la elección de la mejor combinación de parámetros de proceso, dado que se manifestó positivamente en los resultados de fatiga. No obstante, es evidente la necesidad de realizar trabajo adicional utilizando técnicas de inteligencia artificial alternativas, por ejemplo: redes neuronales, algoritmos genéticos o una combinación de ellas, con la finalidad de comparativamente establecer resultados en forma individual por cada técnica o híbridos, para no únicamente cubrir aspectos como en la presente aplicación, sino extender la posibilidad de explorar aplicaciones en otras áreas de ingeniería.

## Referencias

1. Suresh, S.: *Fatigue of Materials*. Second ed. Cambridge, University Press (1998)
2. Miller, K.J.: Materials science perspective of metal fatigue resistance. *Materials Science and Technology*. 9(6), pp. 453–462 (1993)
3. Chadwick, D.J., Ghanbari, S., Bahr, D.F., Sangid, M.D.: Crack incubation in shot peened AA7050 and mechanism for fatigue enhancement. *Fatigue and Fracture of Engineering Materials and Structures*, 41(1), pp. 71–83 (2018)
4. Croccolo, D., Cristofolini, L., Bandini, M., Freddi, A.: Fatigue strength of shot-peened nitrided steel: optimization of process parameters by means of design of the experiment. *Fatigue and Fracture of Engineering Materials and Structures*. 25(7), pp. 695 (2002)
5. Khany, S.E., Moyeed, M.A., Siddiqui, M.S., Ahmed, G.M.S., Baig, M.M.A.: An Experimental Study of the Effect of Shot Peening on the Low Carbon Steel and Identification of Optimal Process Parameters. *Materials Today: Proceedings*, 2(4), pp. 3363–3370 (2015)
6. Unal, O.: Optimization of shot peening parameters by response surface methodology. *Surface and Coatings Technology*, 305, pp. 99–109 (2016)
7. George, P.M., Pillai, N., Shah, N.: Optimization of shot peening parameters using Taguchi technique. *Journal of Materials Processing Technology*, pp. 153–154, pp. 925–930 (2004)
8. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2), pp. 111–127 (1997)

9. Lin, J.L., Wang, K.S., Yan, B.H., Tarng, Y.S.: Optimization of the electrical discharge machining process based on the Taguchi method with fuzzy logics. *Journal of Materials Processing Technology*, 102(1), pp. 48–55 (2000)
10. Gupta, A., Singh, H., Aggarwal, A.: Taguchi-fuzzy multi output optimization (MOO) in high speed CNC turning of AISI P-20 tool steel. *Expert Systems with Applications*, 38(6), pp. 6822–6828 (2011)
11. Kamble, P.D., Waghmare, A.C., Askhedkar, R.D., Sahare, S.B.: Multi Objective Optimization of Turning AISI 4340 Steel Considering Spindle Vibration Using Taguchi-Fuzzy Inference System. *Materials Today: Proceedings*, 2(4), pp. 3318–3326 (2015)
12. Moayyedean, M., Abhary, K., Marian, R.: Optimization of injection molding process based on fuzzy quality evaluation and Taguchi experimental design. *CIRP Journal of Manufacturing Science and Technology* (2018)
13. Zadeh, L.: Fuzzy sets. *Information and Control*, 8, pp. 338–353 (1965)
14. Dubois, D., Prade, H.: *Fuzzy sets and systems: Theory and Applications*. Academic Press (1980)
15. Kosko, B.: Fuzzy systems as universal approximators. 43(11), pp. 1329–1333 (1994)
16. Mandani: Application of fuzzy logic to approximate reasoning using linguistic synthesis, *IEEE Transactions on Computers* (1977)
17. Nam, Y.S., Jeon, U., Yoon, H.K., Shin, B.C., Byun, J.H.: Use of response surface methodology for shot peening process optimization of an aircraft structural part. *The International Journal of Advanced Manufacturing Technology*, 87(9), pp. 2967–2981 (2016)
18. Rodopoulos, C.A., Curtis, S.A., De los Rios, E.R., Solís-Romero, J.: Optimisation of the fatigue resistance of 2024-T351 aluminium alloys by controlled shot peening-methodology, results and analysis. *International Journal of Fatigue*, 26(8), pp. 849–856 (2004)

# Identificación y control difuso en el diagnóstico de fallas para sistemas una entrada-una salida: aplicado a la dirección de un robot tractor

Raúl Cortes-Gutiérrez<sup>1,2</sup>, Julio C. Ramos-Fernández<sup>1,2</sup>,  
Juan David Padre-Nonthe<sup>1</sup>, Marco A. Márquez-Vera<sup>1,2</sup>,  
Filiberto Muñoz-Palacios<sup>1,2</sup>

<sup>1</sup> Universidad Politécnica de Pachuca,  
Posgrado Maestría en Mecatrónica, Zempoala,  
México

<sup>2</sup> Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos,  
México

c.g-raul@hotmail.com, jramos@upp.edu.mx, jhonsnail.23blue@hotmail.com,  
marquez@upp.edu.mx, mupafi@upp.edu.mx

**Resumen.** Un problema en los sistemas mecatrónicos que trabajan en ambientes agresivos, por ejemplo la agricultura, es la degradación parcial o total de los dispositivos o sistemas de medición, que cierran los lazos de control. Si esto sucede en la dirección de un robot tractor, y no está establecida de manera electrónica y con algoritmos adecuados la supervisión en la operación del sistema, se pueden causar perjuicios al propio sistema y civiles. En el presente trabajo, se propone una solución técnica y científica, que consiste en el modelado, identificación y control difuso para sistemas no lineales del tipo una entrada-una salida. Se emplea la teoría de submodelos difusos del tipo Takagi-Sugeno (TS), para identificar y sintonizar una ley de control. Se determina el error residual del sistema en tiempo real contra el modelo difuso en lazo cerrado, para detectar las fallas. Se provocan diferentes fallas simulando la degradación del sensor de posición en una dirección del tipo Ackerman de un Robot Tractor (RT) experimental. Los resultados en simulación y del sistema real que se muestran, indican que los sistemas que se describen con la técnica de TS, son una buena opción para resolver este tipo de problemas en el diagnóstico de fallas en la mecatrónica.

**Palabras clave:** diagnóstico de fallas, identificación y control difuso, robot tractor.

## Identification and Fuzzy Control in the Diagnosis of Faults for Systems One Input-One Output: Applied to the Steering of a Tractor Robot

**Abstract.** A trouble mechatronic systems that work in aggressive environments, for example agriculture, is the partial or total degradation

of measuring devices or systems, which close the control loops. If this happens in the steering of a tractor robot, and it is not established electronically and with adequate algorithms the supervision in the operation of the system, can cause damages to the system itself and civilians. In the present paper, a technical and scientific solution is proposed, which consists of modeling, identification and fuzzy control for non-linear systems of an input-output type. The theory of fuzzy submodels of the Takagi-Sugeno (TS) type is used to identify and tune a control law. The residual error of the system in real time is determined against the fuzzy model in closed loop, to detect the faults. Different faults are caused by simulating the degradation of the position sensor in a steering of the Ackerman type of an experimental Robot Tractor (RT). The simulation and real system results show that the systems described with the TS technique are a good option to solve this type of problem in the diagnosis of faults in mechatronics.

**Palabras clave:** fault diagnosis, identification and fuzzy control, robot tractor.

## 1. Introducción

En la actualidad existen dispositivos y sistemas que por su complejidad demandan un alto grado de confiabilidad en sistemas de seguridad para detección y aislamiento de fallas, cuando algún sistema opera en condiciones de fallo las consecuencias provocadas pueden poner en riesgo la integridad del sistema, daños a los operarios, pérdidas económicas.

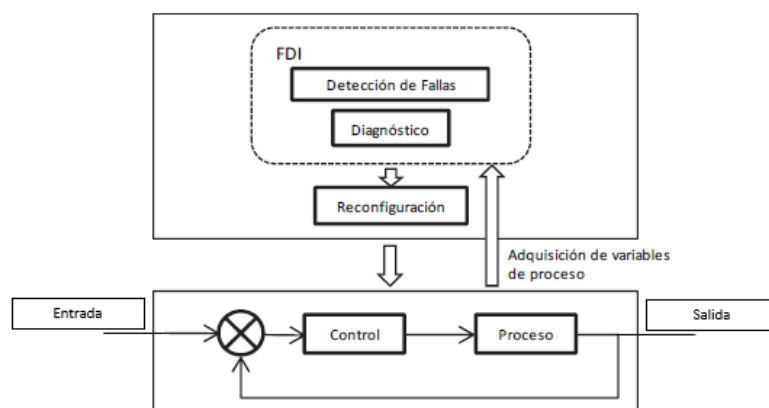
Los sistemas de detección de fallas han evolucionado adaptando las técnicas de control disponibles, la teoría de control clásico ha sido empleada en la detección y aislamiento de fallas como se muestra en [1], en dichas técnicas se concentran incertidumbres debido a dinámicas y parámetros no contemplados al modelar un sistema, esta es una limitación general al usar técnicas de control clásico, además las técnicas basadas en el control clásico, no permiten hacer clasificación de un conjunto de fallas, otra desventaja de este método es que si una falla no se modela específicamente no hay garantía que el sistema pueda detectarlo, es por eso que una buena opción es el uso de técnicas basadas en modelos que utilizan la informática industrial y la Inteligencia Artificial.

Como se evidencia en [2], las redes neuronales y la lógica difusa que permiten realizar clasificación y reconocimiento de patrones en los diferentes modos de operación, han sido ampliamente utilizadas en el desarrollo de sistemas de detección y diagnóstico de fallas, debido a sus habilidades de aprendizaje, puesto que se tratan de aproximadores universales como se muestra en [3]. El uso de técnicas de inteligencia artificial para abordar la detección y aislamiento de fallas constituye en la actualidad un campo activo de investigación.

En [4] se presentan trabajos que emplean arquitecturas de modelos basados en formulaciones diferenciales y analíticos, y técnicas de control difuso, considerando las alternativas planteadas para la detección y diagnóstico de fallas.



Existen algunas características deseables como lo son: la rapidez de adaptación ante cambios en las salidas del sistema, la capacidad de clasificación en distintas fallas, las cuales es posible comparar el desempeño de las estrategias como se detallan en [5]. Una arquitectura conocida para el diagnóstico y detección de fallas que se emplea en el presente trabajo de investigación se muestra en la Figura 1.



**Fig. 1.** Esquema de detección y diagnóstico de fallas.

El esquema presentado en la Figura. 1 se compone de dos bloques principales:

- a).- La parte inferior representa el control en lazo cerrado del proceso.
- b).- El bloque superior es el sistema de detección y diagnóstico de fallas, dicho bloque se observa como se obtiene una adquisición de variables de proceso, las cuales son comparadas dentro del sistema de detección de fallas, para así reconfigurar la variable de salida del control en caso de existir algún tipo de falla.

Las técnicas de control difuso presentadas en este artículo forman una parte esencial en la etapa de diseño y control del sistema de dirección en el robot tractor, hay dos tipos principales de controladores basados en lógica difusa como se muestra en [7], que se describen a continuación:

a) Mamdani: Utiliza reglas tipo si-entonces (if-else). Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y el consecuente, en un sistema difuso tipo Mamdani tanto el antecedente como el consecuente de las reglas están definidos por expresiones lingüísticas, como se muestra en [6].

b) Takagi-Sugeno: Las reglas de la base de conocimiento de un sistema Takagi-Sugeno son diferentes en su estructura de los consecuentes a las de los sistemas Mamdani, el consecuente de estas reglas ya no es una etiqueta lingüística, es una función analítica en forma lineal o no lineal, por ejemplo la descripción de espacio de estado, como puede observarse en [6].

En [8, 9] se presentan diferentes métodos de control difuso diseñados según el modelo identificado en cada sistema. El diseño de un controlador usando el enfoque basado en reglas articula tres fases de implementación. Estas son: la fase de adquisición de conocimiento o entrenamiento, la fase de desarrollo del modelo y la fase de prueba del modelo. Los modelos difusos T-S representan la dinámica global de sistemas reales no lineales, como se expone en [10-13]. El modelo tipo T-S de un sistema no lineal, generalmente se basa en un conjunto de modelos lineales locales que se fusionan suavemente con la estructura del modelo difuso, un enfoque natural y directo es diseñar un controlador local para cada modelo local del proceso. Esta idea se conoce como compensación distribuida paralela (PDC), que es un aporte teórico fundamental de los científicos Tanaka y Sugeno principalmente, en la teoría de identificación y control difuso, con la prueba de estabilidad donde se emplea la desigualdad cuadrática de Lyapunov para sintonizar un controlador que hace estable al sistema, como se muestra en [14].

Los avances en las capacidades de diseño mecánico, las tecnologías de detección de fallas, la electrónica y los algoritmos de planificación y control han llevado a la posibilidad de realizar operaciones de campo basadas en plataformas robóticas autónomas. Los trabajos de investigación en el campo del desarrollo de un robot tractor, han aportado a la fecha un importante número de técnicas y métodos para el desarrollo y diseño de elementos en sistemas integrales, en [15] se muestra el desarrollo científico y tecnológico de un sistema para el trabajo agrícola, que consiste en 3 RT, que realizan labores agrícolas perfectamente sincronizados con tecnología RTK-GPS, con un error de posicionamiento de 2cm, éste es un trabajo de agricultura de precisión El desarrollo de la autonomía en vehículos autónomos terrestres ha evolucionado en una corriente exponencialmente creciente, en [16] se desarrolló un robot tractor con especificaciones para la agricultura de precisión.

Para resolver el problema de la escasez de mano de obra agrícola debido al envejecimiento ya que según el INEGI la edad promedio de los trabajadores agrícolas es de 41.7 años [17], el campo enfrenta un serio problema de envejecimiento, cerca de 60 por ciento de los trabajadores agrícolas tiene más de 60 años, lo que los mantiene debajo de la línea de bienestar y repercute en la baja productividad e incluso el abandono de la tierra según datos de [18]. Con el estudio del robot tractor que se desarrolla en este trabajo de investigación, se impulsa el desarrollo científico y tecnológico de la agricultura de precisión en México.

Una plataforma de experimentación adecuadamente instrumentada, permite el aprovechamiento de todos los recursos teóricos en el diseño y desarrollo de estrategias de control para los actuadores que forman parte del sistema de estudio, en [19] se presenta el trabajo previo a este estudio donde se muestra la instrumentación y automatización de un RT. Se presenta en [20], el diseño y desarrollo del control de trayectoria para un RT, donde se emplea un regulador autoajutable. Se realiza en [21] un trabajo de investigación que se centra en las características, medidas de rendimiento, tareas y operaciones agrícolas donde

la aplicación de robots en operaciones de la agricultura ha sido ampliamente demostrada.

El sistema de dirección es uno de los componentes con mayor relevancia en el seguimiento de trayectorias y la autonomía de un RT, el sistema de dirección ha sido objeto de estudio para realizar y diseñar estrategias de control a través de distintas técnicas. En [22] se presenta un control para el torque de una dirección asistida para el seguimiento del par para volantes asistidos que se emplean en la industria automotriz.

Elementos externos tales como sensores, actuadores, mecanismos son empleados en los sistemas de dirección, con el fin de realizar un control robusto de dicho sistema. En [23] se desarrolló, construyó y probó un mecanismo intercambiable sobre la dirección de vehículos todoterreno y maquinaria agrícola.

Existe un sistema de dirección en el cual básicamente se realiza el control de un servomotor, en [24] se presenta la composición básica de este sistema de dirección asistida eléctrica que presenta soluciones de diseño razonables de hardware y métodos de corrección de un controlador comercial. Una solución particular que se emplea en la industria de vehículos autónomos todoterreno es utilizar un actuador electrohidráulico ( $E/H$ ) para implementar el control de la dirección, en [26] se reporta el diseño y la validación de un controlador de dirección electrohidráulica a través de una combinación de identificación del sistema, simulación del modelo y pruebas de campo.

Los resultados de la prueba se usaron para identificar las características no lineales y dinámicas del sistema de dirección electrohidráulico original, el modelo de identificación del sistema se usó para desarrollar un controlador preliminar, que se simuló en Matlab antes de comenzar la prueba de vehículo a gran escala. Un reporte de diagnóstico de fallas en el sistema de dirección se ha documentado en [26], donde se presenta una plataforma experimental a la cual se le aplica una unidad de control para un actuador que acciona el sistema de dirección.

## **2. Servomecanismo de dirección**

En el presente trabajo de investigación se automatizó un tractor Jhon Deere serie D100, la dirección del RT funciona mediante un sistema de piñón y cremallera, el sistema de dirección de piñón y cremallera de alta resistencia mecánica, dotado de rodamientos proporciona un control sencillo y preciso durante toda la vida útil del tractor, el radio de giro reducido de 40,6 cm, ofrece una buena maniobrabilidad. Las dos barras de dirección equilibran las cargas de dirección y reducen los errores en la dirección.

Formado por una rueda dentada (piñón) y un engranaje plano (cremallera), al girar el piñón desplaza la cremallera en línea recta. Transmite el movimiento y lo transforma de rectilíneo a circular y viceversa.

El trabajo que se presenta en este artículo, consta de un servomecanismo con un motor de corriente directa, acoplado mecánicamente al eje del volante como se aprecia en Figura 2, para mayor detalle ver [19].



**Fig. 2.** Mecanismo acoplado motor-eje de transmisión.

La medición del ángulo de salida en la dirección del vehículo, se realizó a través del acoplamiento de un potenciómetro multivuelta concéntrico al eje longitudinal del volante de la dirección, por medio de dicho dispositivo electrónico es posible registrar un voltaje directamente relacionado con la posición final del ángulo de salida de las llantas directrices del vehículo.

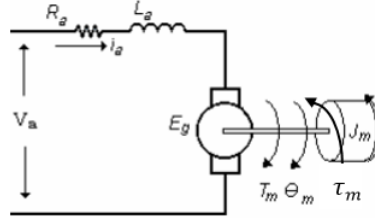
Se realizaron experimentos de posicionamiento de la dirección en lazo abierto, para determinar un modelo del sensor de posición o el ángulo de salida del modelo de la dirección Ackerman, en específico del RT en estudio. La ecuación (1) muestra el resultado del modelo obtenido:

$$\theta(v) = (34.9839)v - 12.8257, \quad (1)$$

donde  $\theta(v)$  es el ángulo de salida en la dirección del vehículo,  $v$  es el voltaje de salida del amplificador de instrumentación.

### 2.1. Motor de CD

El modelo de motor de CD empleado es de la marca SHINANO KENSHI, con un torque de  $1.5Nm$ ,  $24VCD$ ,  $300RPM$ ,  $1/12hp$ . El circuito eléctrico de la armadura y el diagrama de cuerpo libre del rotor se muestran en la Figura 3.



**Fig. 3.** Circuito equivalente del motor de CD.

En el circuito equivalente se introducen las siguientes variables:

- $R_a$  resistencia de armadura.
- $L_a$  inductancia de armadura.
- $v_a$  voltaje de entrada equivalente en este caso al *PWM* suministrado.
- $i_a$  corriente que circula por el circuito.
- $E_g$  representa la fuerza contraelectromotriz.
- $T_m$  par producido.
- $\theta_m$  desplazamiento angular.
- $J_m$  momento de inercia.
- $\tau_m$  fricción en el sistema.

$$V_a(s) = R_a I_a(s) + L_a I_a(s) \cdot s + E_g(s). \quad (2)$$

El voltaje de entrada  $V_a(s)$ , es equivalente a una señal de control *PWM*( $s$ ) se reescribe la ecuación (2) y se obtiene (3):

$$PWM(s) = R_a I_a(s) + L_a I_a(s) \cdot s + E_g(s). \quad (3)$$

Se obtiene la función de transferencia de velocidad  $\omega(s)$ , con respecto de la entrada  $v_a$ , la ecuación de la función de transferencia se muestra a continuación (4):

$$\frac{\omega(s)}{PWM(s)} = \frac{k_a}{J_m L_a s^2 + (J_m R_a + \tau_m L_a) s + (k_b k_a + \tau_m R_a)}. \quad (4)$$

En el presente trabajo, el interés es modelar la posición angular, se sabe que la posición se expresa como (5):

$$\theta(s) s = \omega(s). \quad (5)$$

Se despeja  $\theta(s)$  de la ecuación (5) y se sustituye  $\omega(s)$  de la ecuación (4), para obtener la función de transferencia de la posición  $\theta(s)$  con respecto al voltaje *PWM* de entrada, y se obtiene (6):

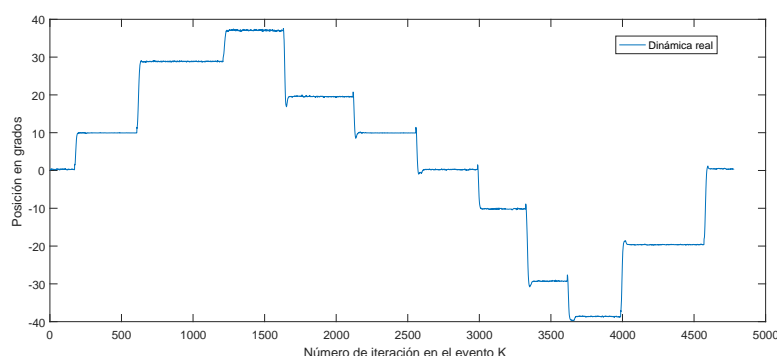
$$\frac{\theta(s)}{PWM(s)} = \frac{k_a}{J_m L_a s^3 + (J_m R_a + \tau_m L_a) s^2 + (k_b k_a + \tau_m R_a) s}. \quad (6)$$

La ecuación (6) indica la relación que existe entre la posición angular del motor y la señal de excitación.

### 3. Parametrización del sistema

Se realizaron experimentos de la manipulación del ángulo de salida de las ruedas directrices del RT, con señales conocidas y acotadas en el dominio del funcionamiento normal de la dirección del RT. Los resultados de la medición entrada-salida, se almacenaron en una base de datos, con un periodo de muestreo de  $20ms$ . En la Figura 4, se muestran datos experimentales de la medición de la posición del ángulo de salida en las ruedas del RT, la respuesta a la variable de entrada ( $PWM$ ) se ilustra en la Figura 5.

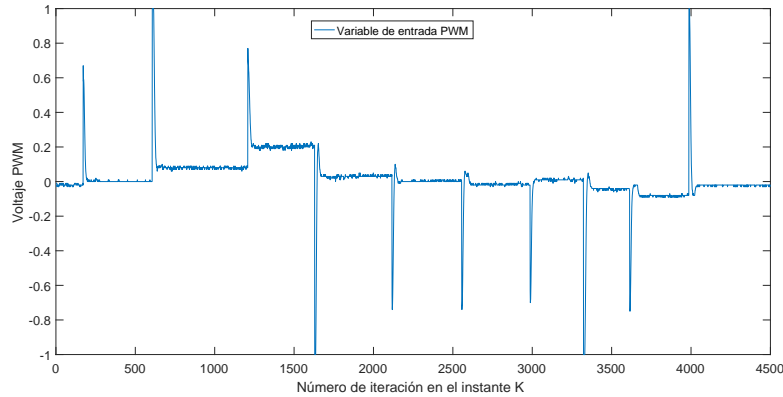
Los datos mostrados en las Figuras 4 y 5, se emplearon para realizar el aprendizaje del modelo difuso que se presenta en este trabajo. Los experimentos realizados se muestran en [19], donde se empleo un modelo a partir de la técnica ANFIS, del cual se obtuvo un controlador que permitió realizar una primer prueba experimental para poder estudiar la dinámica de operación del servomecanismo de la dirección, y así obtener los datos necesarios para realizar el modelo que se presenta en este artículo, la identificación de operación de los parámetros se realizó en lazo cerrado.



**Fig. 4.** Dinámicas de la variable de salida para obtener un modelo por aprendizaje.

Mediante la herramienta *ident* de Matlab y tomando como referencia el comportamiento real del sistema, se obtuvieron dos submodelos en representación de función transferencia en forma discreta que describen dos dinámicas del sistema; se obtuvo una primera función de transferencia para todas las dinámicas de movimiento con un ángulo positivo y por otra parte se obtuvo una segunda función de transferencia para las dinámicas de movimiento con ángulo negativo.

Los modelos resultantes de cada una de las dinámicas del sistema se obtuvieron en el dominio de  $z$ , con una entrada  $PWM$ , y a la salida la posición



**Fig. 5.** Variable de la entrada PWM, para modelar el sistema entrada-salida.

en grados del sistema ( $POS$ ), a continuación se presentan las dos funciones de transferencia resultantes.

La función de transferencia que describe todas las dinámicas positivas se observa en la ecuación (7):

$$\frac{POS(z)}{PWM(z)} = \frac{0.3314z + 0.2832}{z^2 - 1.622z + 0.6239}. \quad (7)$$

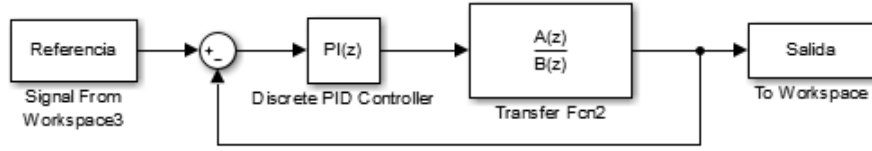
La representación de la función de transferencia para las dinámicas negativas del sistema se presenta en la ecuación (8):

$$\frac{POS(z)}{PWM(z)} = \frac{0.1329z + 0.1205}{z^2 - 1.743z + 0.7438}. \quad (8)$$

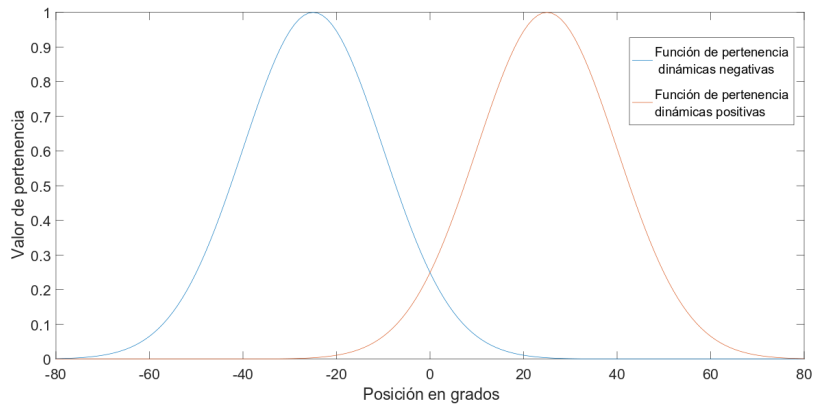
#### 4. Estrategia de control

Se propone un control de tipo proporcional más integral ( $PI$ ) para cada una de las funciones de transferencia obtenidas, que representan la dinámica de operación para que el volante del RT gire a izquierda o derecha. Por medio de la herramienta PID-Tune de Matlab, se sintonizó el controlador  $PI$ , en la Figura 6 se muestra el diagrama a bloques del controlador y la planta con retroalimentación unitaria para la función de transferencia de ambas dinámicas de operación, del mismo modo se realizó este proceso para el submodelo con las dinámicas negativas del sistema.

Se proponen dos reglas difusas del tipo TS con funciones de pertenencia del tipo Gaussiana; los centros de cada submodelo se definieron en 25 y -25 respectivamente para el submodelo con dinámicas positivas y el submodelo con dinámicas negativas, la distribución normal  $\sigma$  se definió con un valor de 10 para cada uno de los submodelos como se aprecia en la Figura 7.



**Fig. 6.** Diagrama a bloques para ambos submodelos lineales en lazo cerrado.



**Fig. 7.** Funciones de pertenencia para el modelo y controlador difuso.

El valor de activación  $\beta$  en cada submodelo se presenta en las siguientes dos ecuaciones (9, 10):

$$\beta_1 = e^{-\frac{(pos-c_1)^2}{2(\sigma_1)^2}}, \quad (9)$$

$$\beta_2 = e^{-\frac{(pos-c_2)^2}{2(\sigma_2)^2}}. \quad (10)$$

Para obtener un controlador difuso PDC, se aplica la técnica de T-S a los dos submodelos difusos que se describen a continuación, con la estructura de reglas difusas. Se nombró  $f_1$  al submodelo con dinámicas positivas y  $f_2$  al submodelo con dinámicas negativas:

$$R_1 : \text{ Si } pos(k) \text{ esta en positivo entonces } \\ f_1 = 1.622 \cdot pos(k) - 0.6239 \cdot pos(k-1) + 0.3314 \cdot U_{Gm} + 0.2832 \cdot U_{Gm}(k-1),$$

$$R_2 : \text{ Si } pos(k) \text{ esta en negativo entonces } \\ f_2 = 1.743 \cdot pos(k) - 0.7438 \cdot pos(k-1) + 0.1329 \cdot U_{Gm} + 0.1205 \cdot U_{Gm}(k-1).$$



Para el cálculo de la posición  $pos(k+1)$  en el modelo aproximado, la defusificación que se emplea es pesos ponderados, se presenta en la siguiente ecuación:

$$pos(k+1) = \frac{\sum_{i=1}^2 \beta_i f_i}{\sum_{i=1}^2 \beta_i}. \quad (11)$$

## 5. Modelo difuso

Es posible diseñar un modelo difuso que represente el comportamiento global del sistema real con sus dinámicas reales, la ley de control difusa  $U_{Gm}$  global, se obtiene a partir de la agregación de los submodelos difusos determinada con el método de varicentro en la siguiente ecuación (12):

$$U_{Gm} = \frac{\sum_{i=1}^R \beta_i ((Kp_i + Ki_i) e_{modelo}(k) - Kp_i e_{modelo}(k-1) + U_{Gm}(k-1))}{\sum_{i=1}^R \beta_i}, \quad (12)$$

donde:

- $\beta_i$  es el valor de disparo de la  $i$ -ésima regla,
- $Kp_i$  es la  $i$ -ésima ganancia proporcional del  $i$ -ésimo controlador,
- $Ki_i$  es la  $i$ -ésima ganancia integrativa del  $i$ -ésimo controlador,
- $e(k)$  es el error en lazo cerrado, de la posición en el  $k$ -ésimo evento:  $e = referencia - posicionreal$ ,
- $e(k-1)$  es el error de la posición en un evento anterior  $k$ .

En la Figura 8 se presenta un diagrama a bloques del control en lazo cerrado y el modelo difuso operando en paralelo.

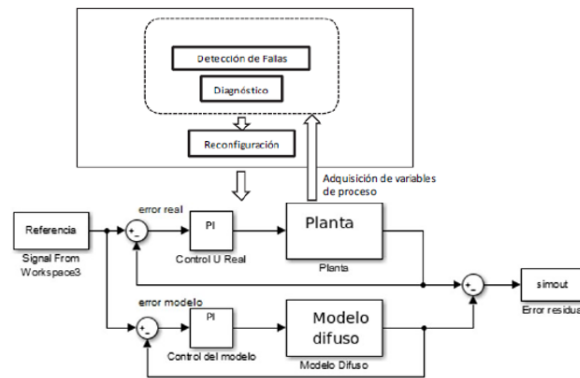


Fig. 8. Esquema de detección de fallas con el error residual.

## 6. Control PDC

A continuación se presenta la base de reglas para el controlador PDC compuesto por cada submodelo:

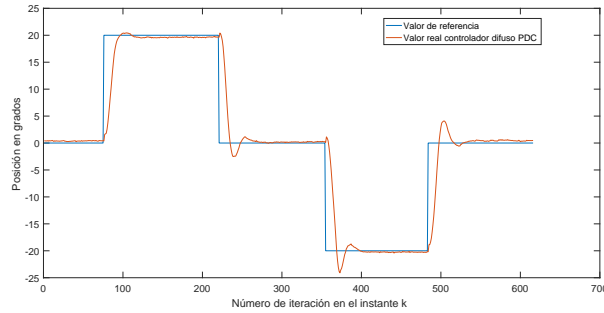
$$R_1 : \text{ Si } pos(k) \text{ esta en angulo positivo entonces} \\ u_1 = ((Kp_1 + Ki_1) e(k) - Kp_1 e(k-1) + U_{GC}(k-1)),$$

$$R_2 : \text{ Si } pos(k) \text{ esta en angulo negativo entonces} \\ u_2 = ((Kp_2 + Ki_2) e(k) - Kp_2 e(k-1) + U_{GC}(k-1)),$$

Se generó una  $U_{GC}$  global de control, a partir de hacer un mapeo en las posición real contra cada función de pertenencia, se obtiene la variable de entrada  $U_{GC}$  en cada iteración, a continuación se presenta la ecuación (13) que representa el valor de  $U_{GC}$ :

$$U_{GC} = \frac{\sum_{i=1}^R \beta_i ((Kp_i + Ki_i) e(k) - Kp_i e(k-1) + U_{GC}(k-1))}{\sum_{i=1}^R \beta_i}. \quad (13)$$

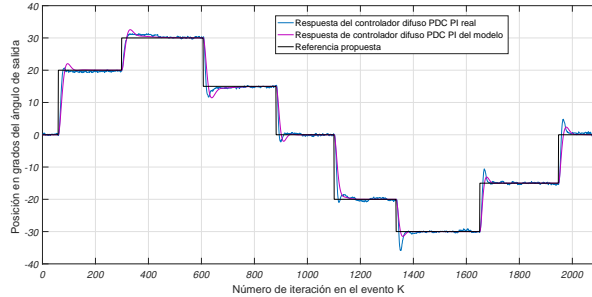
En la Figura 9, se observa el resultado en tiempo real de la regulación del sistema de dirección, para llegar a una posición con el controlador difuso PDC PI.



**Fig. 9.** Regulación de posición utilizando el controlador difuso PDC PI.

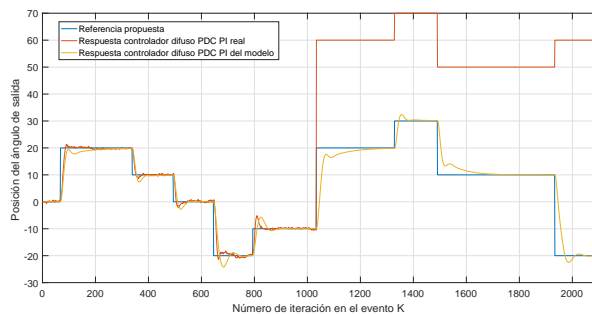
## 7. Resultados

En la Figura 10, se muestra el resultado en tiempo real de la regulación de la planta y el modelo ambos con el control PDC PI de forma paralela.



**Fig. 10.** Regulación de la dirección en tiempo real con control difuso PDC PI real y control difuso del modelo con PDC PI.

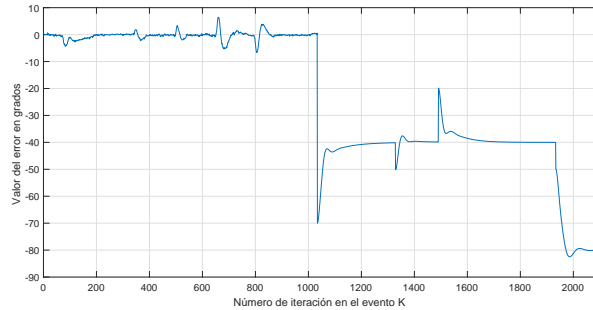
En la Figura 11, se muestra el comportamiento en tiempo real del diagnóstico, detección y aislamiento de la falla inducida en el sistema, la cual consiste en adicionar una elevación de 40 unidades a la posición medida, se observa como la salida real del sistema se eleva 40 unidades como se simula la falla en tiempo real, al mismo tiempo se observa como el modelo difuso en lazo cerrado con el control PDC sigue la referencia propuesta a pesar de que no existe un valor real del sensor de posición.



**Fig. 11.** Diagnóstico y aislamiento de la falla producida en el sistema real.

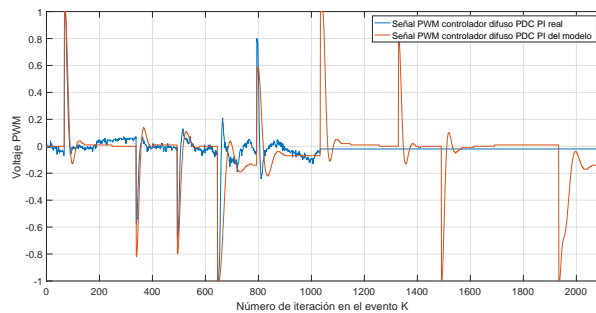
En la Figura 12, se observa con el error residual sobrepasa una banda establecida de .5 grados, discriminando el evento transitorio y considerando un error en estado estacionario absoluto de .5 grados entre el error real y el error del modelo, el tiempo transitorio no sobrepasa el límite establecido.

En la Figura 13, se muestra como la variable de excitación *PWM* del control real es aislada y se lleva a cero, en este instante el control del modelo difuso conmuta para excitar a la planta en tiempo real, para llevarla a las posicio-



**Fig. 12.** Comportamiento del error residual ante la falla.

nes deseadas y completar la tarea, aún con el sensor de posición en modo de degradación.



**Fig. 13.** Señal de control PDC PI con sensor y señal de control PDC PI del modelo difuso.

## 8. Conclusiones y trabajo futuro

Se diseñó un modelo difuso por aprendizaje de datos reales y con consecuentes en función transferencia del sistema de dirección del RT. Fue posible proponer y desarrollar una estrategia de control que permitió la correcta regulación de la posición en tiempo real, por medio de un controlador difuso PDC PI, los controladores de cada submodelo se sintonizaron con la herramienta de Matlab PID-Tune, se obtuvieron resultados en tiempo real de la regulación del ángulo de salida de la dirección del RT, con error de posición promedio menor a 0.5 grados.

La ley de control diseñada permitió regular la salida del sistema de dirección en la posición del ángulo de salida. Se produjo una falla al sistema provocando

una elevación de lectura en el sensor de posición de ángulo de salida, y se constató que el error residual de la variable medida con el modelo aproximado en lazo cerrado, es útil para conmutar a modo de degradación del sensor, con regulación del sistema en las posiciones deseadas por el usuario.

Se realizó un control difuso PDC PI para el modelo difuso diseñado, que trabaja de forma paralela al control con el sensor en tiempo real. El modelo obtenido es la base para la detección de fallas del sistema en estudio, por medio del error residual entre la salida real del sistema y la salida que determina el modelo difuso, esto hace posible la detección y el aislamiento de una falla de sensor.

### 8.1. Trabajo futuro

El modelo y control difuso que se ilustra en este trabajo, da la posibilidad de realizar una nueva investigación para establecer criterios de diagnóstico y detección de fallas, bajo premisas como la energía disipada y el tiempo de restablecimiento del sistema en diferentes condiciones reales de terrenos en la navegación del robot tractor.

**Agradecimientos.** El autor Raúl Cortes-Gutiérrez es becario CONACyT No. 611156. Los autores agradecen al CONACyT y al Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos No. 295536, por el apoyo económico en este proyecto.

## Referencias

1. Venkatasubramanian, V., Rengaswamy, R., Yin K., Kavuri S.: A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27(3), pp. 293–311. DOI: 10.1016/S0098-1354(02)00160-6. (2003)
2. Angeli, C., Chatzinikolaou, A., Patsi, S.: On-Line Fault Detection Techniques for Technical Systems: A Survey. *International Journal of Computer Science and Applications*, I(1), pp. 12–30 (2004)
3. Wang, L.X.: Fuzzy systems are universal approximators. In: *Proceeding of the 1st IEEE Conference on Fuzzy Systems*, pp. 1163–1170 (1992)
4. Venkatasubramanian, V., Rengaswamy, V., Kavuri, S., Yin, K.: A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3), pp. 313–326. DOI: 10.1016/S0098-1354(02)00161-8 (2003)
5. Venkatasubramanian, V., Rengaswamy, V., Kavuri, S., Yin, K.: A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers and Chemical Engineering*, 27(3), pp. 327–346. DOI: 10.1016/S0098-1354(02)00162-X (2003)
6. Blej, M., Azizi, M.: Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Fuzzy Real Time Scheduling . *International Journal of Applied Engineering Research*, 11(22), pp. 11071–11075 (2016)

7. Abdullah, J.H., Al-Gizi, M.W., Mustafa, M., Alsaedi, A., Zreen, N.: Fuzzy Control System Review. Department of electrical International Journal of Scientific and Engineering Research, 4(1) (2013)
8. Mann, G.K.I., Bao-Gang, H., Member, R., Gosine, G.: Analysis of Direct Action Fuzzy PID Controller Structures. Transactions on systems, man, and cybernetics—part b: Cybernetics, 29(3) (1999)
9. Aceves-Lopez, A., Aguilar-Martin, J.: A simplified version of Mamdani's Fuzzy Controller: The Natural Logic Controller. Fuzzy Syst, 14(1), pp.16–30 (2006)
10. Radu-Emil, P., Hans, H.: A survey on industrial applications of fuzzy control. Computers in Industry (2010)
11. Babuska, R., Verbruggen, H.B.: A survey on industrial applications of fuzzy control. Control Engineering Practice, 4(11), pp. 1593–1606 (1996)
12. Sala, A., Guerra, T.M., Babuska, R.: Perspectives of fuzzy systems and control. Fuzzy Sets and Systems, 156(3), pp. 432–444 (2005)
13. Cao, S.G., Rees, N.W., Feng, G.: Analysis and design of fuzzy control systems using dynamic fuzzy global models. Fuzzy Sets and Systems, 75(1), pp. 47–62 (1995)
14. Tanaka, K., Wang, H.O.: Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach. John Wiley and Sons (2001)
15. Noboru, N.: Japan Agriculture based on Robot Farming System. Hokkaido University, 9(9), pp. 065–8589 (2001)
16. Liangliang, Ch.Z., Liangliang, Y., Noboru, N.: Development of a Human-driven tractor following a Robot System. Laboratory of Vehicle Robotics, Graduate School of Agriculture (2014)
17. INEGI: Estadísticas a propósito del... día del Trabajador Agrícola (15 de Mayo). (2015)
18. MILENIO: Política milenio.com. <http://www.milenio.com/politica>. (2018)
19. Padre-Nonthé, J.D.: Instrumentación de los servomecanismos de un tractor agrícola. Tesis de Maestría, Posgrado en Mecatrónica PNPC Universidad Politécnica de Pachuca (2018)
20. Fernandez, B., Herrera, P.J., Cerrada, J.A.: Self-tuning regulator for a tractor with varying speed and hitch forces. Computers and Electronics in Agriculture, 145, pp. 282–288 (2018)
21. Avital, B., Vigneault, C.: Agricultural robots for field operations. Part 2: Operations and systems. Institute of Agricultural Engineering, Agricultural Research Organization, The Volcani Center (2017)
22. Leea, D., Yia, K., Changb, S., Leeb, B., Jangc, B.: Robust steering-assist torque control of electric-power-assisted-steering systems for target steering wheel torque tracking. School of Mechanical and Aerospace Engineering, Seoul National University (2018)
23. Aghkhani, M.H., Abbaspour-Fard, M.H.: Automatic off-road vehicle steering system with a surface laid cable: Concept and preliminary tests. Research Center for Agriculture Machinery (RCAM), College of Agriculture, Ferdowsi University of Mashhad (2009)
24. Chang-Shenga, F., Yan-linga, G.: Design of the Auto Electric Power Steering System Controller. Northeast Forestry University (2012)
25. Zhang, Q., Wu, D., Reid, J.F., Benson, E.R.: Model recognition and validation for an off-road vehicle electrohydraulic steering controller. Agricultural Engineering Department, University of Illinois at Urbana–Champaign (2002)
26. Zhang, Q., Wu, D., Reid, J.F., Benson, E.R.: Fault Diagnosis and Safety Design of Automated Steering Controller and Electronic Control Unit (ECU) for Steering Actuator. California PATH Research Report (2005)

## Regulación de voltaje de un convertidor *buck-boost* mediante su modelo difuso inverso

Nadia S. Zúñiga-Peña<sup>1</sup>, Marco A. Márquez-Vera<sup>1</sup>, Julio C. Ramos-Fernández<sup>1</sup>,  
Luis F. Cerecero-Natale<sup>2</sup>, Filiberto Muñoz-Palacios<sup>1</sup>

<sup>1</sup> Universidad Politécnica de Pachuca,  
Departamento de Mecatrónica, Pachuca,  
México

<sup>2</sup> Universidad Tecnológica de Riviera Maya,  
Ingeniería en Mantenimiento, Quintana Roo,  
México

nasamzp@gmail.com, {marquez, jramos, mupafi}@upp.edu.mx  
luis.cerecero@utrivieramaya.edu.mx

**Resumen.** Se muestra la aplicación del modelo inverso difuso para la regulación de voltaje en un convertidor buck-boost, los convertidores dc-dc entregan un voltaje constante incluso cuando hay fluctuación en el voltaje de entrada o en la resistencia de carga, de manera que cuando existen perturbaciones de este tipo, el controlador debe regular el ciclo de trabajo para mantener el punto de operación especificado. Para lograr sobreponerse a la perturbaciones externas, el modelo difuso debe adaptar las reglas difusas que caracterizan al convertidor, con este fin se empleó un filtro digital y el algoritmo de mínimos cuadrados, dado que la frecuencia de operación es de 20KHz se requiere que la evaluación de la señal de control se obtenga rápidamente, por lo cual el modelo difuso debe emplear pocas reglas. Con el fin de cumplir con lo descrito, se propone usar un sistema difuso tipo Sugeno para evitar la fase de retirar la parte difusa del controlador, lo cual es a su vez conveniente si se emplean mínimos cuadrados recursivos para adaptar el modelo difuso.

**Palabras clave:** Lógica difusa, convertidor buck-boost, modelo inverso difuso.

## Voltage regulation for a buck-boost converter by using its inverse fuzzy model

**Abstract.** It is shown an application of an inverse fuzzy model for voltage regulation in a buck-boost converter, the dc-dc converter supply a constant voltage even with fluctuations in the power supply or disturbances in the resistance load, thus when disturbances are presented, the controller must change

the duty cycle to regulate the converter's output. To overcome the changes in the power input or load output, the fuzzy model can update the fuzzy rules with a digital filter and least squares, for real applications the control signal must be computed fast enough due the working frequency of 20KHz, for this reason this fuzzy model has only eight fuzzy rules. Also it is proposed a Sugeno fuzzy inference system to avoid the defuzzification, and this kind of fuzzy systems can be updated with the least squares proposed as well.

**Keywords:** Fuzzy logic, buck-boost converter, inverse fuzzy model.

## 1. Introducción

Los convertidores dc-dc son empleados en muchos aparatos eléctricos como hornos de microondas, computadoras, robots, aeronaves, equipo de telecomunicaciones etc. [1]. Un convertidos dc-dc emplea una fuente de voltaje y un sistema de conmutación que conecta la entrada de voltaje a un inductor, cada vez que el interruptor cambia de posición, el inductor entrega la energía que almacenó a la carga, la cual es comúnmente vista como una resistencia, en paralelo a la carga se conecta un capacitor de modo que el convertidor puede llegar a interpretarse como un filtro LC.

Existen diferentes tipos de convertidores, el tipo buck es un reductor de voltaje, donde el voltaje de salida es menor al voltaje de entrada, éste es el convertidor más sencillo de diseñar. Las variaciones en el voltaje de alimentación del convertidor se ven atenuadas en el voltaje de salida [2], dada su característica de reducir el voltaje de salida, la corriente que entrega es mayor a la corriente de alimentación del convertidor; además, si el interruptor se mantiene todo el tiempo encendido, el voltaje de salida sería igual al voltaje de entrada.

Para lograr una eficiencia cercana al 90% la frecuencia de operación se propone de al menos 20KHz [4] debido a que frecuencias menores pueden ser detectadas por el oído humano lo cual resulta desagradable, la intensidad del ruido depende de la corriente eléctrica en la bobina, por ejemplo el zumbido en algunos reguladores de voltaje es debido a su operación a 50 o 60 Hz, pero frecuencias mayores a 6.3 KHz pueden ser más molestas [5]; de cualquier modo, al usar frecuencias aún mayores, el filtro LC puede emplear valores de inductancia y capacitancia más pequeños pero para una corriente eléctrica cercana a lo máximo que pueda conducir el interruptor puede ocasionar que el interruptor no funcione correctamente, ya que de emplear un transistor BJT el efecto sería que el transistor se quedaría saturado debido al tiempo de apagado y a la energía almacenada debido a su capacitancia interna, por otro lado al emplear un MOSFET incremental como interruptor se notarían problemas para encender debido al tiempo requerido para formar el canal interno [3].

En el caso particular de un convertidor buck, si el interruptor se mantiene encendido, el voltaje de salida sería igual al voltaje de entrada, pero al emplear un convertidor tipo boost o uno buck-boost, si el ciclo de trabajo es cercano al 100%, la corriente en la bobina, en el interruptor y en el diodo crecería hasta sobrecalentar alguno de estos elementos, en la subsección 2.1 se describen los componentes propuestos para el convertidor a modelar.



El convertidor boost emplea otra configuración en la ubicación del interruptor y la bobina, ahora el voltaje de salida es mayor al de entrada, en esta configuración resulta más complicado reducir el efecto de perturbaciones externas, ya que se amplifica su efecto [4], en este tipo de convertidor, si el interruptor se mantiene abierto, el voltaje de salida sería igual al de entrada, pero si se mantiene cerrado, la corriente en el inductor sería muy alta ya que se tendría un corto-circuito en dc [6].

En este trabajo, se realiza el control de un convertidor buck-boost en el cual el voltaje de salida puede ser de mayor o menor amplitud que la entrada, presentando el mismo problema que el convertidor boost si el interruptor se mantiene cerrado, algo interesante del buck-boost es que el voltaje de salida presenta polaridad inversa a la entrada [7]. En el caso de un convertidor buck el circuito puede interpretarse como un filtro LC aplicado a una señal de PWM [3], pero para los otros dos la bobina es la encargada de entregar la corriente necesaria a la carga por lo que es importante evitar saturar su núcleo, para este fin, el inductor se carga cuando el interruptor está cerrado y entrega su energía a la carga cuando se abre el switch, el rizo de voltaje es filtrado por el capacitor en paralelo con la carga.

Existen algunas modificaciones para mejorar este tipo de convertidores, por ejemplo, existe el convertidor tipo Cuk que emplea una bobina más en la salida, logrando así un mejor filtrado en el voltaje y un comportamiento similar al del buck-boost, otro convertidor es el SEPIC que emplea dos inductores y dos capacitores, es similar al Cuk pero no invierte la polaridad de la salida [1].

Para controlar el voltaje de salida, manteniendo los parámetros del circuito constantes, se calcula el ciclo de trabajo para determinar los tiempos en que el interruptor de cierra, este diseño se realiza en lazo abierto sin retroalimentar la salida. En trabajo de Guldemir [8] se emplearon modos deslizantes en un convertidor buck-boost, esta aplicación fue debida a la naturaleza discontinua del modelo

Para simular al convertidor buck-boost se requiere evaluar el modelo suponiendo una frecuencia de conmutación de varios KHz, para esto se empleó el modelo propuesto por Sira-Ramírez y Silva-Ortigoza [9], uno similar fue presentado por Pawlak [10] empleando también modos deslizantes para controlar el voltaje. En trabajo de Mahery y Babaei [11] se presenta un modelo matemático y el análisis del transitorio y estado estacionario de convertidores de potencia.

En las simulaciones mostradas en la literatura se emplean interruptores ideales, además del interruptor se emplea un diodo para restringir la dirección de la corriente eléctrica también asumido ideal [9, 10, 16]. En este trabajo, se considera el efecto de dispositivos reales. El control difuso propuesto emplea el valor de la corriente eléctrica en la bobina para calcular el PWM a aplicar en el interruptor. Para medir la corriente eléctrica, se usa una resistencia en serie con el inductor, dado que la eficiencia se reduce por usar dicha resistencia [12] en la Fig. 6 de [1] se presenta como varía el voltaje de salida ante diferentes valores de la resistencia en serie con la bobina, en su Fig. 7 se muestra la eficiencia que suele ser entre 70 y 90% [1].

Dado que se trata de un sistema no-lineal invariante en el tiempo [12], no es suficiente con emplear teoría de control lineal [8], una alternativa son los modos deslizantes dado el modelo discontinuo del convertidor [13]. Una aplicación de este tipo de control en un convertidor tipo boost la presentó Guldemir [14]; algunos conceptos sobre la estabilidad de este controlador se dan en la Proposición 2 de [9], un

estudio sobre el convertidor buck se presentó por Guldemir [15], así como por Reddy y Banakar [16].

Dado que pueden presentarse perturbaciones externas, el modelo podría tratarse como variante en el tiempo, aunque los modos deslizantes presentan robustez y garantizan la estabilidad del sistema en lazo cerrado [17], existen también aplicaciones de control adaptable para mejorar el control ante perturbaciones o variación paramétrica de los componentes por calentamiento [18]. No obstante, no es posible determinar cómo cambiarán los parámetros o que perturbaciones se presentarán, para esto se puede emplear lógica difusa, la cual puede operar ante incertidumbre [2, 19].

Dadas las ventajas mencionadas de la lógica difusa se propone un controlador difuso a partir de la inversión del modelo del convertidor, este tipo de control puede operar en lazo abierto cancelando las dinámicas del sistema, pero si ocurren perturbaciones, se requiere ajustar el modelo difuso, lo cual se logra mediante mínimos cuadrados recursivo. El principal objetivo de este trabajo es usar información sobre el voltaje de salida y la corriente en la bobina para generar la señal de control a partir del modelo difuso aproximado adaptado en línea.

Este trabajo está organizado de la siguiente manera, primero en la sección Materiales y Métodos se presenta el modelo matemático del convertidor buck-boost y el modelo difuso obtenido a partir de mínimos cuadrados teniendo como entradas el voltaje del capacitor, el voltaje de la resistencia en serie con la bobina para estimar su corriente, así como una señal analógica que representa el PWM en el interruptor. A continuación, se presentan los Resultados donde se aprecian las simulaciones cuando se tienen perturbaciones en la fuente de alimentación y en la resistencia de carga. Finalmente, se presentan las Conclusiones obtenidas.

## 2. Materiales y métodos

En esta sección se presentan el diseño del convertidor y algunos conceptos sobre lógica difusa, también se muestra el diagrama del convertidor y el esquema de control empleado.

### 2.1. Convertidor buck-boost

Para la implementación del convertidor, se propuso emplear un transistor MOSFET incremental de canal N como interruptor, donde un valor lógico 1 represente al interruptor cerrado evitando una lógica negada, el transistor es el IRFZ44 el cual puede operar a una frecuencia de hasta 2MHz soportando 60V entre la fuente y el drenaje ( $V_{DS}$ ) con una corriente de conducción de 36A ( $I_D$ ) a una temperatura de 25°C o 50A a 100°C, su resistencia interna es de  $0.028\Omega$  ( $R_{DSon}$ ) [20].

El diodo a su vez debe soportar la corriente de operación y la frecuencia del PWM empleado. La carga considerada es de  $10\Omega$  ( $R_L$ ) el voltaje de alimentación considerado es de 12 V ( $V_{in}$ ) sin perturbaciones y el voltaje de salida deseado es de 24V, por lo que la corriente en el diodo de al menos 2.4 A, para esto se propone usar el diodo RURD4120S9A [21], estos valores propuestos son debido a que las fuentes de alimentación en el laboratorio entregan hasta 5A, para el convertidor la corriente de entrada de alrededor de  $2(24V)/10\Omega=4.8A$ .

En este trabajo se propone emplear una frecuencia de conmutación de 20 KHz ya que no se desea un ruido audible para el usuario, la resistencia para conocer la corriente en el inductor debe ser lo suficientemente pequeña para no afectar la eficiencia del convertidor, según los comentarios de Erickson [1] si la resistencia es 100 menor a la resistencia de carga, la eficiencia puede superar el 90%, por lo que se propone una resistencia de precisión de  $0.1\Omega$ .

En la Fig. 1 se muestra el diagrama del convertidor buck-boost, el diagrama fue dibujado con el software libre Livewire, en el caso en que el voltaje de salida debe variar respecto al tiempo se dice que el convertidor es dc-ac [12].

La ganancia de voltaje del convertidor está dada por la ecuación 1:

$$\frac{v_{out}}{v_{in}} = -\frac{D}{1-D}, \quad (1)$$

donde  $D$  representa el ciclo de trabajo del interruptor siendo un número entre cero y uno. El Inductor se calcula para tener siembre conducción de corriente en el circuito sin que el diodo recorte el flujo, para esto el rizo de corriente no debe tener valores negativos evitando así reducir la eficiencia en un modo de conducción continua.

Asumiendo una eficiencia de 100%, y despreciando las no-linealidades de los componentes, ya que a 20KHz la deformación de los pulsos de PWM no es de consideración en los componentes electrónicos, se obtiene que:

$$\frac{v_{out}^2}{R_L} = v_{in} I_L D, \quad (2)$$

donde  $I_L$  representa el valor medio de la corriente eléctrica en el inductor y  $R_L$  es la resistencia de carga, con esta idea y considerando la frecuencia de 20KHz y un ciclo de trabajo mínimo de  $D=0.2$ , ahora se obtiene:

$$I_L = \frac{v_{in} D}{R_L (1-D)^2}. \quad (3)$$

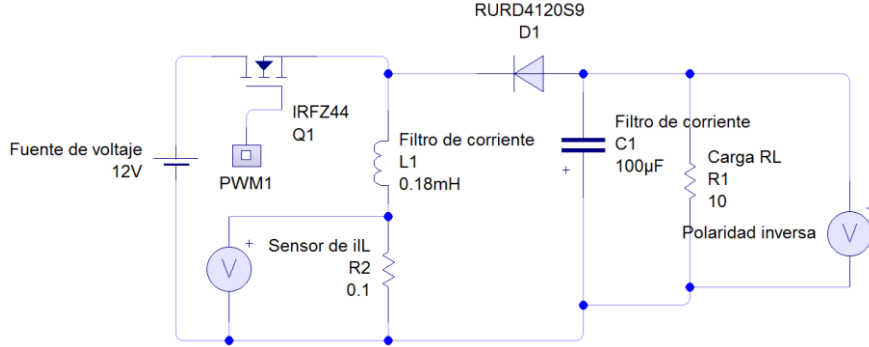
El ciclo de trabajo mínimo  $D=0.2$  se eligió para tener un voltaje en la salida mayor a cero para construir un modelo difuso a partir de datos de encendido, en este caso el valor mínimo de salida de voltaje sería alrededor de -3V asumiendo un diodo ideal en el circuito recordando que el convertidor empleado invierte la polaridad en la salida.

El rizo de la corriente  $\Delta i_L$  debe tener como valor mínimo cero para tener un estado de conducción continua siendo:

$$i_{Lmin} = I_L - \frac{\Delta i_L}{2} = \frac{v_{in} D}{R_L (1-D)^2} - \frac{v_{in} D T}{2L}, \quad (4)$$

donde  $T$  es el periodo de conmutación; para evitar valores negativos en la corriente con el fin de no recortar energía y no reducir reduciendo la eficiencia, se propone que siempre  $i_L > 0A$  [9], de modo que la inductancia mínima está dada por la ecuación 5:

$$L_{min} \geq \frac{R_L (1-D)^2}{2f} = 16mH, \quad (5)$$



**Fig. 1.** Convertidor Buck-Boost.

para esto, se propuso emplear un inductor de 18mH. El voltaje de salida presenta un rizo  $\Delta v_{out}$  filtrado por el capacitor, siendo el rizo:

$$\Delta v_{out} = \frac{v_{out} DT}{R_L C}. \quad (6)$$

De este modo, la capacitancia mínima requerida para filtrar el voltaje con un rizo menor al 1% se obtiene mediante la ecuación 7:

$$C_{min} \geq \frac{D}{R_L f \frac{\Delta v_{out}}{v_{out}}} = 100 \mu F, \quad (7)$$

el cual es un valor comercial. El modelo matemático fue presentado en [8, 9], donde la señal  $u$  toma los valores 0 ó 1, indicando la conmutación entre dos modelos, uno para el caso cuando el transistor MOSFET está encendido (saturación) y para el caso de apagado (corte), dada la conmutación del transistor, éste no opera en la región lineal y la potencia que disipa es despreciable. El modelo matemático tiene dos variables de estado, la corriente eléctrica en la bobina  $i_L(t)$  y el voltaje en el capacitor  $v_C(t)$ , la señal de control está denominada como  $u_{in}(t)$  y la salida es el voltaje en la carga, el mismo que en el capacitor, el modelo está dado por la ecuación 8:

$$\begin{aligned} \frac{di_L(t)}{dt} &= (1-u) \frac{v_C(t) - 0.5}{L} + \frac{v_{out} - 0.1i_L(t)}{L} u_{in}, \\ \frac{dv_C(t)}{dt} &= -(1-u) \frac{i_L(t)}{C} - \frac{v_C(t)}{R_L C}, \end{aligned} \quad (8)$$

donde 0.5V es un voltaje que se resta debido al voltaje de conducción del diodo propuesto RURD4120S9A, otro diodo que puede emplearse es el BYW29-200 que logra operar a 28 KHz conduciendo 5A con una caída de voltaje de 0.8V o inclusive un diodo Schottky como el STPS1L40-Y que se emplea en la industria automotriz debido a su rápida respuesta;  $0.1i_L(t)$  representa el voltaje en la Resistencia propuesta como medio para estimar la corriente en la bobina, en la Fig. 1 se presenta el esquema del convertidor.

## 2.2. Control difuso

El control difuso emergió como una herramienta para tratar con información incompleta o imperfecta, la idea es trabajar con variables lingüísticas que son más fáciles de interpretar por algún operador que las variables numéricas [22]. Los sistemas difusos pueden ser interpretados como un filtro que absorbe los cambios en las mediciones debido a errores de modelado o incertidumbre paramétrica. Un control difuso es un mapeo de las entradas a las salidas del sistema mediante relaciones lingüísticas, ampliamente utilizado, incluso en electrodomésticos como hornos para arroz, lavavajillas o cámaras fotográficas, ahora la palabra fuzzy llega a ser sinónimo de “amigable con el usuario” o “buen desempeño”. Para construir el control difuso se empleó el modelo difuso obtenido con mediciones de un experimento anterior, la señal de entrada al convertidor es pseudo-aleatoria, misma que al ser comparada con una señal triangular para obtener una señal de PWM, tomando los valores  $i_L(t)$  y  $v_C(t)$  se construye un modelo reglas difusas de la forma:

$$R^i : \text{Si } i_L(kT) \text{ es } A^i \text{ y } v_C(kT) \text{ es } B_i \text{ y } u_c(kT) \text{ es } C_i, \\ \text{entonces } v_C((k+1)T) = \theta^i \quad (9)$$

donde  $A_i$ ,  $B_i$  y  $C_i$  son conjuntos difusos,  $\theta_i$  es un consecuente escalar para la  $i$ -ésima regla difusa dado que se trata de un sistema difuso tipo Sugeno, al tener una partición difusa con dos funciones de pertenencia para cada premisa, se cuenta con un modelo monotónico con lo que se puede garantizar la invertibilidad del modelo para obtener un controlador [23], además de tener la menor cantidad de reglas difusas posible [24],  $2^3 = 8$  reglas en esta aplicación.

El vector de estado está denotado por  $X$  y la salida deseada es  $y$ , los consecuentes escalares de las reglas difusas son obtenidos mediante mínimos cuadrados [23] usando la ecuación 10:

$$\theta_i = (X^T w_i X)^{-1} X^T w_i y_i, \quad (10)$$

donde  $X$  está formada por los valores de la corriente eléctrica en la bobina, el voltaje en el capacitor y el PWM de entrada,  $w_i$  es una matriz diagonal cuyos elementos son las pertenencias normalizadas obtenidas con las funciones de pertenencia.

Ahora, con los consecuentes de las reglas, se evalúa el modelo difuso con las mismas condiciones iniciales que el convertidor:

Para  $kT=0$

$$R^i : \text{Si } i_L(0) \text{ es } A^i \text{ y } v_C(0) \text{ es } B_i \text{ y } u_{in}(0) \text{ es } C_i, \\ \text{entonces } v_f(1) = \lambda_i^T(0)\theta_i; \quad (11)$$

Para  $kT > 0$

$$R^i : \text{Si } i_L(kT) \text{ es } A^i \text{ y } v_C(kT) \text{ es } B_i \text{ y } u_{in}(kT) \text{ es } C_i, \\ \text{entonces } v_f((k+1)T) = \lambda_i^T(kT)\theta_i,$$

donde  $\lambda_i(kT)$  es la pertenencia normalizada de las tres variables  $i_L(kT)$ ,  $v_C(kT)$  y  $u_c(kT)$ , por lo que se obtienen tres parámetros constantes para los consecuentes teniendo  $v_f$

$((k+1)T) = \lambda_i^T(kT)\theta_i$ , la agregación de las reglas difusas se realiza empleando la ecuación 12:

$$\lambda_i(kT) = \frac{\prod_{j=1}^8 \mu_{A_j}(x_i(kT))}{\sum_{i=1}^8 \prod_{j=1}^8 \mu_{A_j}(x_i(kT))}, \quad (12)$$

En (10)  $X$  representa las variables de estado, un esquema del modelo con las ocho reglas propuestas se muestra en la Fig. 2. La comparación entre la salida del modelo difuso y el voltaje de salida del convertidor se presenta en la Fig. 3 donde la línea azul es llamada "Real converter" denotando la salida real del convertidor, la línea roja es la salida del modelo difuso empleando la misma condición inicial que la del convertidor y es llamada "Fuzzy model".

En la Tabla 1 se presentan los consecuentes  $\theta$  de las reglas para el modelo difuso  $v_c((k+1)T) = f(i_L(kT), v_c(kT), u_{in}(kT))$ .

Con los límites entre  $[0, 1]$ , para el caso del voltaje del capacitor, la saturación está entre  $[-25, 0]$  y los límites de la corriente eléctrica en la bobina son  $[0, 8]$ ; el modelo difuso inverso mapea  $u_{in}(kT) = f^{-1}(x(kT), r((k+1)T))$ , siendo  $r$  la referencia deseada de voltaje, los centros  $C_j$  para calcular la señal de control se obtienen usando la ecuación 13:

$$C_j = \sum_{i=1}^2 \prod_{i=1}^2 \mu_{A_i}(x_i) \theta_i^j. \quad (13)$$

Para que el modelo difuso se monotónico es necesario que  $C_1 < C_2 < C_3 < \dots < C_n$  o  $C_1 > C_2 > C_3 > \dots > C_n$ , cada vez que se calcula los centros [23], al tener únicamente dos de ellos el modelo obtenido es monotónico; ahora la señal de control se calcula con la ecuación 14:

$$u_{in}(kT) = \sum_{j=1}^2 \mu_{C_j}(r(k+1)T) C_j, \quad (14)$$

donde  $C_j$  es el  $j$ -ésimo centro usado para calcular la señal de control. La simulación del control en lazo abierto usando el modelo difuso inverso se muestra en la Fig. 4, se usó un modelo de referencia para suavizar la señal de referencia, para usar un algoritmo de adaptación se requiere que la salida no cambie abruptamente si la referencia tiene discontinuidades [25]. El modelo de referencia es un filtro pasa-bajas dado por la función de transferencia en la ecuación 15:

$$\frac{Y_m(z)}{R(z)} = \frac{0.0198z}{z - 0.9802}, \quad (15)$$

Una forma más sencilla sería calcular el ciclo de trabajo  $D$  a partir de (1), obteniendo así la ecuación 16:

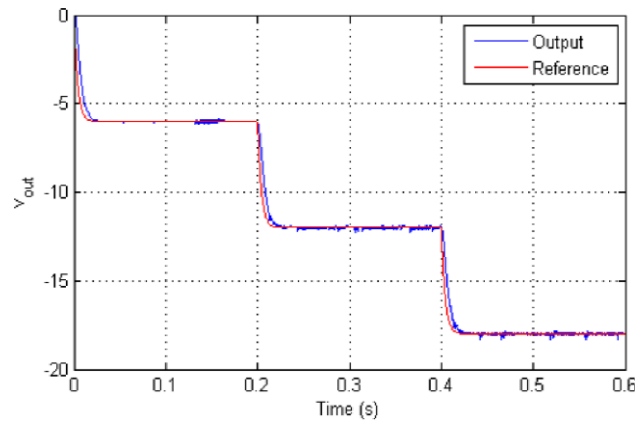


Fig. 4. Control difuso en lazo abierto

Tabla 1. Consecuentes de las reglas.

$X(kT)=(v_c(kT), i_L(kT))$	$u_{in}(kT)$	
	small	big
X1(small, small)	$\theta_1^1=-23.7358$	$\theta_1^2=-24.8571$
X2(small, big)	$\theta_3^1=-30.7124$	$\theta_4^2=-22.6584$
X3(big, small)	$\theta_5^1=-0.1096$	$\theta_6^2=-0.6775$
X4(big, big)	$\theta_7^1=12.4612$	$\theta_8^2=-3.7125$

$$D = \frac{v_{out}}{v_{out} + v_{in}}. \quad (16)$$

Un problema ocurre cuando existen perturbaciones externas modificando la respuesta del convertidor, para contrarrestarlas se puede realizar el control en lazo cerrado con los sensores, para esto se calcula el error entre la salida del convertidor y la del modelo difuso para luego filtrarlo debido a la naturaleza discontinua del sistema; siendo la salida del modelo inverso la diferencia entre la señal de referencia y el error filtrado, además, mediante mínimos cuadrados recursivos se pueden actualizar los consecuentes de la reglas difusas del modelo en línea. Para obtener mejores resultados debe tomarse en cuenta el signo de la retroalimentación debido a que el convertidor buck-boost invierte la polaridad de la salida. Un esquema del controlador en lazo cerrado se muestra en la Fig. 5.

### 3. Resultados y discusión

Se simularon algunas variaciones en la resistencia de carga y en el voltaje de entrada a manera de perturbaciones. El controlador obtenido con el modelo difuso inverso puede operar en lazo abierto, pero ante las perturbaciones simuladas se realizó la adaptación del modelo difuso, se simuló un cambio en RL de 10 a 15Ω cuando el voltaje de referencia (set point) es menor a -8V en magnitud que el voltaje de entrada, de forma

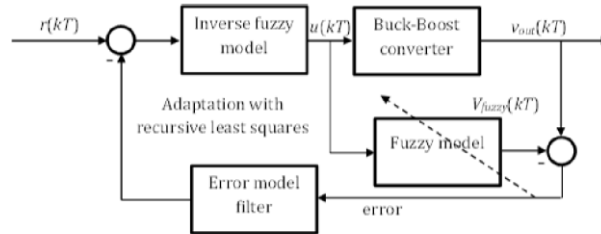


Fig. 5. Esquema de control en lazo cerrado.

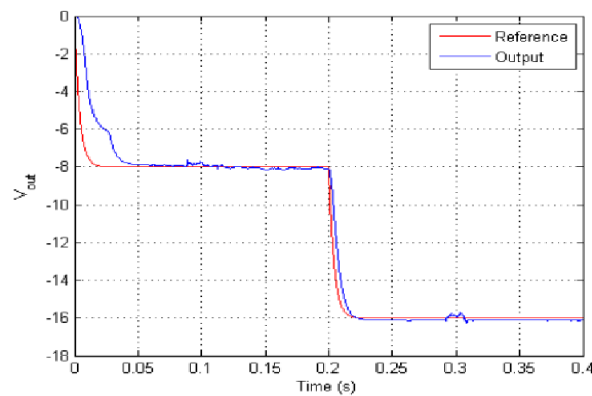


Fig. 6. Control en lazo cerrado con perturbaciones en la carga y voltaje de entrada.

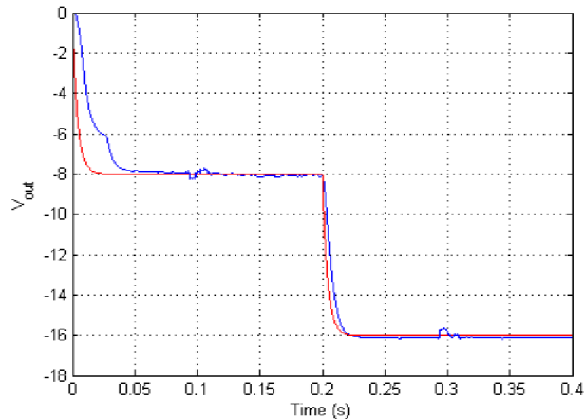


Fig. 7. Control en lazo cerrado con adaptación en los consecuentes de las reglas del modelo.

que el voltaje de salida podría incrementarse, esto se simuló entre los segundos [0.8s, 1.2s] Fig. 6; después se varió la carga de 10 a 5Ω cuando el voltaje deseado es -16V, el cual es más grande en magnitud que el voltaje de entrada, ahora la demanda de corriente en la carga es mayor lo que podría generar que el voltaje de salida sea menor que el esperado, esta perturbación se da en el intervalo [2.8s, 3.2s].

El resultado de emplear el control en lazo cerrado usando el modelo difuso inverso se presenta en la Fig. 7. Para mejorar el desempeño del controlador se emplearon



mínimos cuadrados recursivos con factor de olvido  $\eta$  usando una matriz inicial  $P(0)$  como se ve en la ecuación 17:

$$\theta(kT) = \theta((k-1)T) + \frac{P((k-1)T)\lambda(kT)}{\eta + \lambda^T(kT)P((k-1)T)\lambda(kT)}, \quad (17)$$

siendo:

$$P(kT) = \frac{1}{\eta} \left( P((k-1)T) - \frac{P((k-1)T)\lambda(kT)}{\eta\lambda^T(kT)P((k-1)T)\lambda(kT)} \right). \quad (18)$$

#### 4. Conclusiones

Se presentó un control mediante la inversión del modelo difuso del sistema para regular el voltaje de un convertidor de potencia Buck-Boost, este control tiene una buena respuesta incluso en lazo abierto, pero para reaccionar ante perturbaciones externas que modifiquen la respuesta del sistema se puede hacer la adaptación en línea del modelo difuso antes de invertirlo, de este modo se cierra el lazo de control atenuando los efectos de las perturbaciones o cambios paramétricos que puedan ocurrir en el convertidor, los consecuentes del modelo difuso se actualizan mediante mínimos cuadrados recursivos, siendo un punto interesante de diseño el factor de olvido, propuesto en este trabajo de 0.98 para evitar oscilaciones en el voltaje de salida.

Si bien este tipo de control limita la máxima salida posible del sistema, una mala elección del factor de olvido puede provocar oscilaciones no deseadas, por lo que la robustez es del control en lazo cerrado es un tema futuro a tratar, así como la aplicación de un control predictivo, dado que ya se cuenta con un modelo aproximado del sistema que además puede ser invertido.

Como trabajo futuro, se planea implementar esta ley de control en un convertidor tipo Cuk, el cual es similar al aquí utilizado, pero empleando control predictivo, además se piensa agregar un término integral para reducir aún más el error en estado estacionario.

**Agradecimientos.** Esta aplicación es con fines de regular el voltaje a aplicarse en vehículos autónomos manejados en el Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos, proyecto 295536, al cual se le agradece el apoyo otorgado en la realización de esta implementación.

#### Referencias

1. Erickson, R.W.: Dc-dc power converters. Wiley Encyclopedia of Electrical and Electronics Engineering. DOI: 10.1002 /047134608X. W5808.pub2 (2001)
2. Marquez-Vera, Muñoz-Palacios, F., Farfán-García, J.M.: Fuzzy control type II in dc-dc converters. In: Proceeding of the IEEE Electronics, Robotics and Automotive Conference, pp. 272–276. DOI: 10.1109/CERMA.2012.51 (2012)
3. Rashid, M.H: Power Electronics: Circuits, Devices and Applications (2014)
4. Hart, D.W.: Introduction to Power Electronics (1996)

5. Lawton, B.W.: Damage to human hearing by airborne sound of very high frequency or ultrasonic frequency. Institute of Sound and Vibration Research (2001)
6. Mohan, N., Undeland, T.M., Robbins, W.P.: Power Electronics: Converters, Applications and Design (2009)
7. Ballester, E., Piqué, R.: Electrónica de Potencia: principios fundamentales y estructuras básicas (2012)
8. Guldemir, H.: Modeling and sliding mode control of dc-dc buck-boost converter. In: Proceedings of the International Advanced Technologies Symposium, pp. 475–480 (2011)
9. Sira-Ramírez, H., Silva-Ortigoza, R.: Modelling of dc-to-dc power converters in Control Design Techniques. In: Power Electronics Devices 2, pp. 11–58 (2006)
10. Pawlak, M.: Modeling and analysis of buck-boost dc-dc pulse converter. In: Proceedings of the International PhD Workshop, pp. 137–142 (2010)
11. Mahery, H.M., Babaei, E.: Mathematical modeling of buck-boost converter and investigation of converter elements on transient and steady state responses. Electrical Power and Energy Systems, 44, pp. 949–963 (2013)
12. Kiprianoff, M.: Prime dc-ac buck-boost converter: derivation of mathematical models and evaluation of lumped transmission lines with focus on size and efficiency. Master's thesis, Chalmers, University of Technology (2012)
13. Sira-Ramírez, H.: On the generalized pi sliding mode control of dc to dc power converters: A tutorial. International Journal of Control, 76, pp. 1018–1033 (2003)
14. Guldemir, H.: Sliding mode control of dc-dc boost converter. Journal of Applied Sciences, 5, pp. 588–592 (2005)
15. Guldemir, H.: Study of sliding mode control of dc-dc buck converter. Energy and Power Engineering, 3, pp. 401–406 (2011)
16. Reddy, P.D., Banakar, B.: Sliding mode control technique for dc-dc buck converter with improved performance. International Journal for Research in Applied Science and Engineering Technology, 3, pp. 271–280 (2015)
17. Mattavelli, P., Rosetto, L., Spiazzi, G.: Small-signal analysis of dc-dc converters with sliding mode control. IEEE Transactions on Power Electronics, 12, pp. 96–102 (1997)
18. Nizami, T.K., Mahanta, C.: An intelligent adaptive control of dc-dc buck converters, Journal of the Franklin Institute. 353, pp. 2588–2613 (2016)
19. Solano-Martínez, J., Hissel, D., Péra, M.C.: Type-2 fuzzy logic control of a dc-dc buck converter. (IFAC) Proceeding, 45, pp. 103–108 (2012)
20. Siliconix, V.: Power mosfet irfz44 (2011)
21. Semiconductor, F.: Rurd4120s9a (2010)
22. Ross, T.: Fuzzy Logic with Engineering Applications, John Wiley & Sons (2008)
23. Babuska, R.: Fuzzy modeling for control (1998)
24. Márquez-Vera, M.A., Ramos-Fernández, J.C., Cerecero-Natale, L.F., Lafont, F., Balmat, J.F., Esparza-Villanueva, J.I.: Temperature control in a miso greenhouse by inverting its fuzzy model. Computers and electronics in Agriculture, 124, pp. 168–174 (2016)
25. Passino, K., Yurkovich, S.: Fuzzy Control, Addison Wesley Longman Inc. (1998)
26. Coughlin, R.F., Driscoll, F. F.: Operational Amplifiers and Linear Integrated Circuits (2001)

## **Representación de eventos de ruido ambiental a partir de esquemas preconceptuales y buenas prácticas de educación geoespacial de requisitos**

Claudia Elena Durango-Vanegas, Paola Andrea Noreña-Cardona,  
Carlos Mario Zapata-Jaramillo

Universidad San Buenaventura,  
Universidad Nacional de Colombia,  
Colombia

claudia.durango@usbmed.edu.co, {panorenac, cmzapata}@unal.edu.co

**Resumen.** El ruido ambiental es un evento que genera contaminación acústica y tiene repercusiones nocivas en la calidad de vida, la salud y el comportamiento humano. Los esquemas preconceptuales son modelos de ingeniería de software para representar diferentes dominios, en los que se suelen modelar eventos. Las representaciones actuales del ruido ambiental se basan en modelos de procesos y estructuras de componentes espaciales para generar mapas de ruido en sistemas de información geográfica (SIG). Sin embargo, les falta considerar la especificación de los eventos y las buenas prácticas de educación geoespacial de requisitos. Estas prácticas permiten relacionar objetos geográficos comunes en SIG. En este artículo se propone una representación de eventos de ruido ambiental basada en esquemas preconceptuales y buenas prácticas de educación geoespacial de requisitos. Esta representación mejora la comprensión de los geodatos requeridos para modelar el comportamiento de los eventos de ruido ambiental en la generación mapas de ruido.

**Palabras clave:** eventos de ruido ambiental, buenas prácticas de educación geoespacial de requisitos, esquemas preconceptuales, ingeniería de software, sistemas de información geográfica.

### **Environmental Noise Representation by using Pre-conceptual Schemas and Geospatial Elicitation Requirements Best Practices**

**Abstract.** Environmental noise is an event, which generates noise pollution. Such event causes harmful effects in quality of life, health, and human behavior. Pre-conceptual schemas are software engineering models for representing any domains, in which events are modeled. Current environmental noise

representations are based on process models and spatial component structures for generating noise maps in geographic information systems (GIS). However, such representations lack events specification and geospatial elicitation requirements good practices. Such practices are used for relating common geographic objects in GIS. In this paper, we propose an environmental noise representation by using pre-conceptual schemas and geospatial elicitation requirements good practices. Such representation improves geodata understanding for modeling environmental noise events behavior in noise maps generation.

**Keywords:** environmental noise events, geospatial elicitation requirements good Practices, pre-conceptual schemas, software engineering, geographic information systems.

## **1. Introducción**

El ruido ambiental es un evento que se considera uno de los principales contaminantes acústicos de la salud en los seres vivos. Por ello, diversas entidades gubernamentales a nivel regional, nacional e internacional realizan estudios de medición del ruido ambiental, buscando mejorar el nivel de calidad de vida y el comportamiento de las personas. Algunas instituciones, como la Organización Mundial de la Salud (OMS), buscan regular el ruido ambiental, por ser la principal causa de preocupación en la salud pública, tratando de encontrar un nivel de confort acústico para las personas afectadas. Con este fin, los eventos de ruido ambiental se miden en una determinada zona de estudio. Los resultados obtenidos indican que el ruido ambiental tiene una alta repercusión nociva en el nivel de calidad de vida, el comportamiento y las actividades cotidianas del ser humano [1].

Los esquemas preconceptuales (EP) son modelos de ingeniería de software para representar un dominio y facilitar la comprensión de los analistas y los interesados. Estos esquemas integran características dinámicas y estructurales que permiten dar una vista completa del dominio en un mismo modelo [2]. Por ello, los EP se utilizan para representar eventos involucrados en cualquier dominio.

Estos eventos permiten analizar el comportamiento del sistema mediante el inicio o el fin de los procesos [3,4]. Aquellos eventos que inician procesos se conocen como eventos disparadores [5]; de esta manera, los eventos de ruido ambiental se clasifican como eventos disparadores.

En algunas representaciones del ruido ambiental se utilizan modelos de procesos como diagramas de flujo [6,7,8], diagramas de clases [9] y diagramas de bloques [10,11], para generar mapas de ruido en sistemas de información geográfica (SIG) a partir de objetos geográficos. Otras representaciones se basan en la integración de estructuras de objetos geográficos para generar mapas de ruido a partir de métodos, tales como colección básica de datos [12], interpolación [13], modelado por escenarios [14] y propagación de sonidos [15]. Los mapas de ruido cobran importancia porque se

pueden realizar análisis y simulaciones de eventos de ruido ambiental en diferentes lugares y proponer soluciones ante su ocurrencia [16].

A pesar de que las anteriores propuestas incluyen representaciones de eventos de ruido ambiental y algunos objetos geográficos para generar mapas de ruido en SIG, hace falta representar eventos de ruido ambiental. Además, estos trabajos no tienen en cuenta las buenas prácticas de educación geoespacial de requisitos para realizar mapas de ruido.

Estas buenas prácticas permiten identificar un terreno común de las cosas que se deben atender del proyecto SIG para modelar conceptual, física y lógicamente las entidades y fenómenos geográficos requeridos para generar mapas de ruido.

En este artículo se propone una solución a partir de la representación de eventos de ruido ambiental, tomando como base los esquemas preconceptuales. Esta representación integra elementos de las buenas prácticas de la educación geoespacial de requisitos para obtener el catálogo de objetos geográficos, el modelado de las estructuras de geoalmacenamiento y el modelo de georepresentación en los proyectos SIG para generar mapas de ruido.

La representación propuesta mejora la comprensión del proceso de medición de eventos relacionados con ruido ambiental. Además, la definición de las buenas prácticas ayuda a identificar los elementos mínimos que se requieren en la adquisición de geodatos para generar mapas de ruido según zonas específicas de estudio.

La estructura de este artículo es la siguiente: en la sección 2 se presenta el marco conceptual relacionado con los mapas de ruido ambiental, los esquemas preconceptuales y las buenas prácticas de educación geoespacial de requisitos; en la sección 3 se plantea el problema, identificando trabajos similares; en la sección 4 se propone una solución basada en la representación de eventos de ruido ambiental a partir de esquemas preconceptuales y buenas prácticas de educación geoespacial de requisitos; en la sección 5 se aplica la solución, y en la sección 6 se presentan las conclusiones y el trabajo futuro.

## **2. Marco conceptual**

### **2.1. Eventos de ruido ambiental**

El ruido ambiental es un evento que genera un sonido indeseable que afecta o perjudica a las personas y su entorno, y es una de las principales fuentes de contaminación ambiental en los centros urbanos [13,16]. El crecimiento de la población, la modernización de las actividades cotidianas, el incremento de las industrias y de los medios de transporte son algunas causas del aumento y de la presencia de eventos de ruido ambiental en centros poblados. En los centros urbanos existen diversas fuentes generadoras de contaminación auditiva producto del ruido ambiental, tales como: transporte automotor, construcciones, obras públicas, ruido industrial y ruido propio de establecimientos públicos y de vecindarios. Lo anterior trae como consecuencia un rompimiento del equilibrio natural, generando estrés por el incremento de los niveles permitidos de ruido en las ciudades [17]. Por ello, el ruido

ambiental se considera un evento de alta repercusión en cambios relacionados con la salud, el comportamiento y las actividades cotidianas de las personas [18]. Los eventos de ruido ambiental se clasifican como eventos disparadores, ya que desencadenan procesos y otros eventos [5].

## **2.2. Esquemas preconceptuales**

Los esquemas preconceptuales (EP) son modelos de ingeniería de software para representar cualquier dominio. Los EP integran reglas de la lingüística computacional y de los modelos conceptuales que permiten un acercamiento al lenguaje natural. Este acercamiento facilita el nivel de comprensión de los analistas y los interesados. Los elementos en la notación de los EP permiten una vista completa del dominio, ya que involucran características dinámicas y estructurales en un mismo modelo [2]. Estos elementos se pueden observar en la Figura 1: relaciones (estructural para las clases, dinámica para los procesos y eventual para los eventos); nodos (condicional, concepto, variable independiente, concepto-clase, y operador); enlaces (implicación, conexión, concepto-nota y operación); y aglutinadores (evento, valor-nota, marco y restricción).

Los EP permiten la representación de eventos disparadores y eventos de resultado (aquellos que son producto de la finalización de los procesos) [5]. Estos eventos permiten el análisis del comportamiento del sistema, ya que ellos cambian los estados de los procesos, por lo que, generalmente, se presentan en la vista dinámica o de comportamiento del sistema [3,4].

## **2.3. Buenas prácticas de educación geoespacial de requisitos**

Los sistemas de información geográfica (SIG) son herramientas que permiten adquirir, almacenar, analizar, actualizar y geovisualizar información espacial o geodatos. Los geodatos contienen información espacial y no espacial de un espacio geográfico, asociando la localización y un sistema de coordenadas [19]. Además, son elementos que definen la funcionalidad del SIG y su manipulación permite identificar los productos de trabajo esperados (mapas dinámicos y estáticos). El ciclo de desarrollo de los proyectos SIG involucra la definición de los siguientes elementos: geodatos, geousuarios, software SIG y método SIG.

Las buenas prácticas son procesos aceptados como correctos y efectivos [20]. Por ello, en la literatura se encuentran diversos métodos para planificar proyectos SIG, relacionados con la adquisición, el almacenamiento, el análisis espacial, el mantenimiento, la actualización y la geovisualización de los geodatos. Estos métodos SIG contienen fases, actividades y productos de trabajo que se relacionan con buenas prácticas y en algunos casos se pueden replicar en otros proyectos SIG. Sin embargo, se debe considerar que los proyectos SIG tienen actividades y productos de trabajo propio para su desarrollo y planificación [21], como es el caso de las buenas prácticas de educación geoespacial de requisitos. Estas buenas prácticas se originan en la identificación de prácticas comunes de algunos métodos de desarrollo de proyectos SIG, tales como: la comprensión del problema geoinformático, la identificación de los

requisitos estructurales de los geodatos y la realización de la estructura de almacenamiento y el catálogo de representación de objetos geográficos.

### 3. Planteamiento del problema

Algunas de las propuestas para la medición y representación del ruido ambiental y la generación de mapas de ruido incluyen modelos de procesos, como diagramas de flujo, a partir del proceso de mapeo de ruido tradicional, para representar las fases de medida de los datos [6]; otras propuestas utilizan este mismo diagrama para calcular el mapa de ruido con base en las mediciones de los datos y sus procesos [7,8]. El diagrama de clases [9] se propone para representar objetos geográficos del ruido ambiental a partir de la vista estructural del proyecto SIG; sin embargo, esta representación carece de elementos para representar eventos y las buenas prácticas de educación geoespacial de requisitos. Otras propuestas utilizan los diagramas de bloques [10,11] como modelos de procesos. Estos modelos tienen una vista dinámica del SIG y se utilizan para representar la secuencia y medida de los geodatos en la generación de mapas de ruido.

Otras propuestas utilizan representaciones que se basan en la integración de estructuras de objetos geográficos a partir de métodos como colección básica de datos [12] para generar mapas de ruido ambiental del tráfico de áreas urbanas.

La interpolación [13] también es un método para generar mapas de ruido ambiental a partir de la predicción de valores para generar mapas de ruido. El modelado por escenarios [14] y la propagación de sonidos [15] son métodos en los que se suelen representar los datos que surgen en el mapa de ruido.

Las anteriores propuestas incluyen algunos métodos y objetos geográficos para generar mapas de ruido; sin embargo, estas propuestas carecen de la representación de eventos de ruido ambiental y los eventos que intervienen en el proceso de medición del ruido. Adicionalmente, no se relacionan los procesos con los objetos geográficos para una representación completa de los elementos y relaciones, que permitan el análisis de la información. Tampoco se encuentra una integración del conjunto de buenas prácticas de educación geoespacial de requisitos que complementen los elementos y la información de los sistemas de información geográfica para la generación de mapas de ruido.

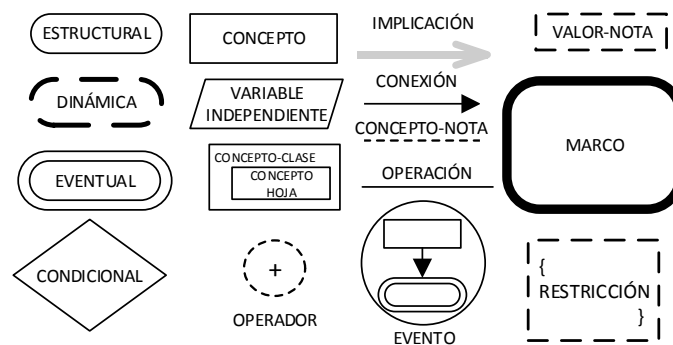


Fig. 1. Notación de los EP.



Fig. 2. Catálogo de representación de los objetos geográficos.

## 4. Propuesta de solución

### 4.1. Definición de actividades en las buenas prácticas de educación geoespacial de requisitos

En esta primera etapa se proponen las siguientes actividades relacionadas con las buenas prácticas de educación geoespacial de requisitos para la representación de eventos de ruido ambiental:

**Catálogo de objetos geográficos.** Identificar los requisitos estructurales de los geodatos. En esta actividad se identifican las entidades, atributos, dominios, relaciones, sistema de coordenadas, escala y límites geográficos, entre otros, para conformar el catálogo de objetos geográficos que debe contener la medición del ruido ambiental. Esta información se obtiene a partir de la interacción entre un geousuario experto en georreferenciación (especialista SIG) y el analista de geodatos; algunos objetos geográficos son zona (zona de interés donde se realiza la medición de los eventos de ruido ambiental), uso del suelo (conjunto de edificaciones en la misma zona) y edificación (espacio en el cual se presenta una construcción). Como información de la edificación se definen el área, el perímetro, la altura y la cantidad de pisos. Estos objetos y su información se deben incluir en la representación de eventos de ruido ambiental. Además, esta información se vincula con el proyecto SIG mediante los programas ArcGIS® para la creación de los mapas de ruido y SoundPLAN® para el análisis de resultados, por lo cual se deben integrar los diferentes elementos y deben ser de fácil comprensión.

**Catálogo de representación de objetos geográficos.** Realizar la estructura de georepresentación de los geodatos. En esta actividad se identifican los símbolos y los patrones de colores asociados con los niveles de ruido ambiental presentes en los mapas de ruido, como se puede observar en la Figura 2. Para el caso de los niveles de ruido, la especificación de colores constituye una escala de valores, tales como:  $\leq 45\text{dB}$  constituye una alerta verde,  $45\text{dB} < 55\text{dB}$  indica una alerta amarilla,  $55\text{dB} < 65\text{dB}$



indica una alerta naranja,  $>65\text{dB}<75\text{dB}$  indica una alerta roja y  $>75\text{dB}$  indica el mayor valor con una alerta azul.

**Modelado de geoalmacenamiento.** Realizar la estructuración de almacenamiento de los geodatos. En esta actividad se modelan los geodatos requeridos según el catálogo de objetos geográficos, buscando mejorar la comprensión de las estructuras de geoalmacenamiento requeridas en los proyectos SIG en la generación de mapas de ruido, mediante el modelado de geoalmacenamiento de las estructuras geográficas requeridas para la generación de los mapas de ruido.

#### **4.2. Representación de eventos de ruido ambiental a partir de esquemas preconceptuales y buenas prácticas de educación geoespacial de requisitos**

En esta segunda etapa se utilizan los EP como propuesta de solución, para facilitar la comprensión de la información entre los geousuarios expertos en georreferenciación y los analistas de geodatos. Por lo tanto, se elabora el esquema preconceptual de la Figura 3 para integrar actividades de las buenas prácticas de educación geoespacial de requisitos que se definen en la Sección 4.2. Este EP involucra los geodatos, los procesos y los eventos que intervienen en la representación de eventos de ruido ambiental para dominios de software científico, que se basan en información geográfica como gestión ambiental. El cumplimiento de esta etapa permite identificar los geodatos que se requieren para generar mapas de ruido según zonas específicas de estudio, en los cuales se centra el modelo.

Para la integración de la actividad de catálogo de objetos geográficos se utilizan las características estructurales de los EP, que permiten observar las relaciones entre conceptos clase y conceptos hoja o atributos mediante la relación estructural *tiene*. De esta manera, los objetos geográficos o geodatos *zona*, *edificación*, *uso del suelo*, y *ruido ambiental* son conceptos-clase. A estos conceptos se suman *ingeniero de sonido* y *medición* como elementos que complementan la representación del dominio en eventos de ruido ambiental. *Zona* tiene dos conceptos hoja *código* y *nombre*. *Zona* se relaciona estructuralmente con *edificación*, *uso del suelo* y *medición*. *Edificación* tiene *código*, *área*, *altura de pisos*, *altura*, *cantidad de pisos*, *perímetro*, *cantidad de ocupantes* y *cantidad de viviendas*. *Uso del suelo* tiene *código* y *nombre*. *Medición* tiene los conceptos hoja *código*, *fecha*, *cantidad de registros*, *suma de valores*, *valor promedio* y *alerta final*. *Ingeniero de sonido* tiene *identificación* y *nombre* y se relaciona estructuralmente con *medición*. *Ruido ambiental* es un concepto-clase que se deriva de los registros que se toman desde un sonómetro y *tiene registro*, *tiempo local*, *valor* y *alerta* (*verde*, *amarilla*, *naranja*, *roja* y *azul*). Los conceptos hoja que se mencionan permiten representar la información de los sistemas de información geográfica que se requieren en la generación de mapas de ruido y en el geoalmacenamiento.

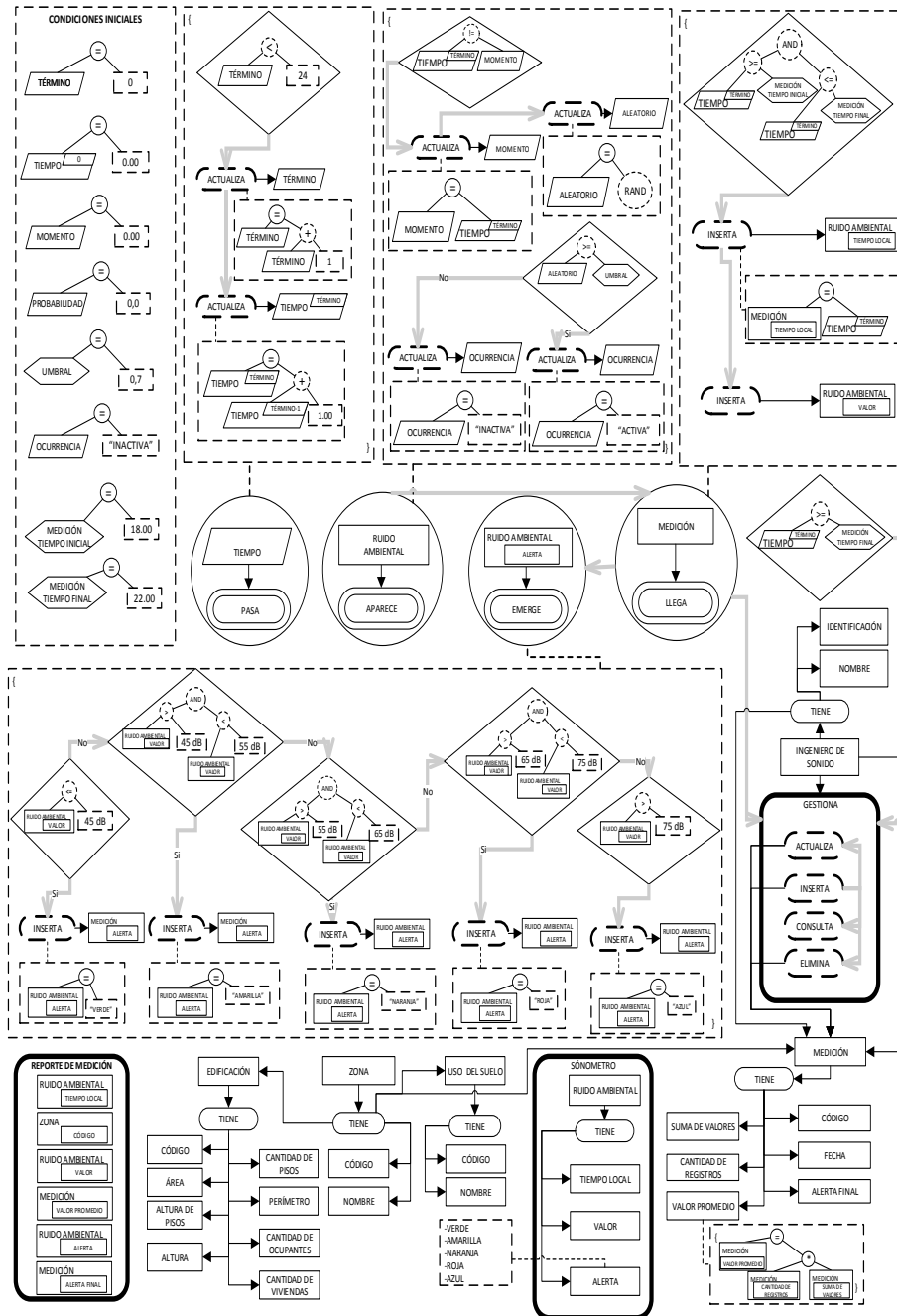
Las características dinámicas de los EP permiten analizar el flujo de los procesos que interactúan con los objetos geográficos mediante relaciones dinámicas y eventuales. Las relaciones dinámicas son operaciones que realiza un objeto animado (persona o rol) como *ingeniero de sonido actualiza medición* y las relaciones eventuales son operaciones automáticas que se realizan sin la intervención de un objeto animado como *ruido ambiental aparece*, por lo que pueden incluir relaciones dinámicas sin

objeto animado en sus restricciones. Para el inicio del flujo se requiere conocer las *condiciones iniciales* del dominio, las cuales se representan mediante una especificación que incluye los valores iniciales de los parámetros (constantes) y las variables independientes o globales. El flujo inicia a partir del evento disparador *tiempo pasa* que tiene como restricción incrementar en '1.00' hora digital, mientras se cumpla la condición  $tiempo < 24$ ; este término permite la ubicación del vector  $tiempo[tiempo]$  incrementando de 1 en 1 hasta 24 como representación de 1 día, con *condiciones iniciales*  $tiempo = 0$  y  $tiempo[tiempo] = 0.00$ .

Durante el tiempo el *ruido ambiental aparece*, este evento tiene una restricción que permite representar la probabilidad de ocurrencia del ruido ambiental, mediante la relación dinámica automática *actualiza aleatorio*, la cual es igual al *operador rand* (función *random* o aleatorio), esta representación se utiliza con base en eventos aleatorios de probabilidad estadística que utilizan generadores de valores aleatorios en procesos de simulación de sistemas, ya que la ocurrencia del ruido ambiental es variable. Si surge un valor que cumpla la condición  $aleatorio \geq umbral$  (parámetro que es igual a '0,7' según las condiciones iniciales, el valor del umbral se estima con base en el conocimiento histórico del proceso) entonces se "activa" la *ocurrencia* con la ejecución de la relación dinámica *actualiza ocurrencia* que tenía previamente como *condición inicial* el valor de "inactiva", si no se cumple la condición tendrá este mismo valor inicial. El *evento ruido ambiental aparece* implica el evento *medición llega*, el cual tiene como restricción la condición si  $tiempo[tiempo] \geq medición tiempo inicial$  y si  $tiempo[tiempo] \leq medición tiempo final$  entonces *inserta el tiempo local* y el *valor del ruido ambiental*.

En la integración de la actividad de catálogo de representación de objetos geográficos, se incluyen los colores de las alertas mediante el uso del evento *alerta de ruido ambiental emerge*, que se dispara con el evento *medición llega*. Este evento contiene una restricción que incluye cinco condiciones que representan los niveles de ruido ambiental, los cuales se comparan con el valor del ruido ambiental que se insertó previamente: si el *valor del ruido ambiental* es  $\leq 45dB$  *inserta alerta del ruido ambiental "verde"*, sino si el *valor del ruido ambiental* es  $45dB < 55dB$  *actualiza alerta del ruido ambiental a "amarilla"*, sino si el *valor* es  $55dB < 65dB$  *actualiza alerta del ruido ambiental a "naranja"*, sino si el *valor* es  $65dB < 75dB$  *actualiza alerta del ruido ambiental a "roja"* sino si el *valor* es  $> 75$  *actualiza alerta del ruido ambiental a "azul"*.

El evento *medición llega* implica la relación dinámica ingeniero de sonido gestiona (inserta, actualiza, elimina y consulta) *medición*, al igual que el evento condicional si  $tiempo[tiempo] \geq medición tiempo final$ . Esto quiere decir, que el ingeniero de sonido debe esperar a que el sonómetro termine de registrar los valores, para gestionar la información. Cuando se cumple la restricción del evento *medición llega* y el evento condicional, el ingeniero de sonido ingresa la información de la medición (código, fecha, código de zona, identificación del ingeniero, cantidad de registros, suma de valores, valor promedio y alerta final). El valor promedio de la medición surge como atributo derivado de la suma de valores y la cantidad de registros, es decir, se calcula al ingresar ambos valores. Al ingresar el registro, el ingeniero también podrá actualizar, eliminar y consultar la información de la medición.



**Fig. 3.** Representación de eventos de ruido ambiental a partir de EP y buenas prácticas de educación geoespacial de requisitos.

La integración de la actividad de modelado de geoalmacenamiento se realiza a partir de las características estructurales de los EP y los eventos, en los cuales se puede observar la información de la representación mediante tablas de geoalmacenamiento que representan la base de datos geoespacial de los sistemas de información geográfica y el funcionamiento de variables independientes utilizando tablas de geodatos.

## 5. Aplicación

Las tablas de la base de datos geoespacial que se integran en la actividad de modelado de geoalmacenamiento permiten la aplicación de los geodatos, clases y eventos del esquema preconceptual de la Figura 3, los conceptos hoja de los geodatos en las tablas permiten guardar los valores que corresponden a cada concepto. El caso de aplicación se realiza en la *zona de la Comuna 11 Laureles-Estadio* de la ciudad de Medellín-Colombia, según el polígono (entidad utilizada para representar superficies) de la Figura 4.

Para iniciar el proceso los ciudadanos que sienten perjuicio en los niveles de ruido pueden llamar a la Subdirección Ambiental del Área Metropolitana para solicitar el servicio de verificación de ruido, según la Resolución 0627 por regulación del Ministerio de Ambiente, Vivienda y Desarrollo Territorial 2006 [22].

El ingeniero de sonido lleva el sonómetro a la *zona* para medir los niveles de ruido ambiental y programa el tiempo inicial y final (parámetros del EP de la Figura 3) en el sonómetro. La simulación del tiempo se define para un día, pero se puede modificar según las condiciones de otras zonas y lugares de un país. Por ello, la Tabla 1 presenta el funcionamiento de la variable independiente *tiempo* y la Tabla 2 de las variables independientes *probabilidad* y *ocurrencia* (con valores aleatorios que se definen en la ocurrencia de un evento estadístico, permiten indicar cuando se activa el evento de ruido ambiental), las cuales no se almacena en la base de datos geoespacial.

Por su parte, las Tablas 3 a 8 se registran como datos de geoalmacenamiento, ya que el concepto-clase *ingeniero de sonido* en la Tabla 3 permite registrar quién está a cargo de la medición.

En la Tabla 4 se registran los geodatos de la zona de estudio de la Comuna 11 Laureles-Estadio, para el caso de *zona* con código “Z701234”. En la Tabla 5 se registran los geodatos de edificaciones que pertenecen a la *zona*, el código “1110001” de *edificación* corresponde a un *Hotel* y el “1111012” corresponde a una *zona Mixta* (residencial y discotecas; véase la Figura 4); *Hotel*, *Mixta* y *residencial* (edificios residenciales) son *usos del suelo* en la Tabla 6.

La Tabla 7 del geodato *ruido ambiental* se llena automáticamente mediante el sonómetro tomando cinco registros desde la *medición inicial* hasta la *final* (estos registros generalmente se toman en una memoria USB y se llevan a la base de datos geoespacial), los cuales varían entre “74dB” y “77dB”. Finalmente, la Tabla 8 permite insertar los datos de la *medición* como la *cantidad de registros* “5”, el *valor promedio* “76dB” y la *alerta final* “Azul”, que indica la máxima alerta. En la Figura 5 se observa el mapa de ruido de la zona de estudio con los datos del geoalmacenamiento.



**Fig. 4.** Zona de estudio “Z701234” Carrera 80 entre Calle 38 y 40. Laureles de Medellín- Colombia.

**Tabla 1.** Funcionamiento de la variable independiente *tiempo*.

Tiempo Pasa	
Término	Tiempo
0	0.00
20	20.00
21	21.00
22	22.00

**Tabla 2.** Funcionamiento de las variables independientes *probabilidad* y *ocurrencia*.

Ruido Ambiental Aparece		
Tiempo	Probabilidad	Ocurrencia
0.00	0,0	INACTIVA
...18.00	0,7	ACTIVA
19.00	0,8	ACTIVA

**Tabla 3.** Tabla del concepto-clase *ingeniero de sonido*.

Ingeniero De Sonido	
Código	Nombre
1893213	Jonathan Ochoa

**Tabla 4.** Tabla del geodato *zona*.

Zona	
Código	Nombre
Z701234	Carrera 80 Entre Calle 38 Y 40

**Tabla 5.** Tabla del geodato *edificación*.

EDIFICACIÓN							
Cód.	Área	Perímetro	Altura	Altura Pisos	Cantidad Pisos	Cantidad Viviendas	Código Zona
1111011	629,3 m <sup>2</sup>	102,0 m	5 m	2,5 m	2	10	Z701234
1111012	3609,3 m <sup>2</sup>	279,1 m	5 m	2,5 m	2	5	Z701234

**Tabla 6.** Tabla del geodato *uso de suelo*.

USO DE SUELO		
Cód.	Nombre	Zona
U1	Hotel	Z701234
U2	Mixta	Z701234
U3	Residencial	Z701234

**Tabla 7.** Tabla del geodato ruido ambiental.

RUIDO AMBIENTAL		
Tiempo Local	Valor	Alerta
18.00	74 dB	Roja
19.00	75 dB	Roja
20.00	77 dB	Azul
21.00	76 dB	Azul
22.00	77 dB	Azul

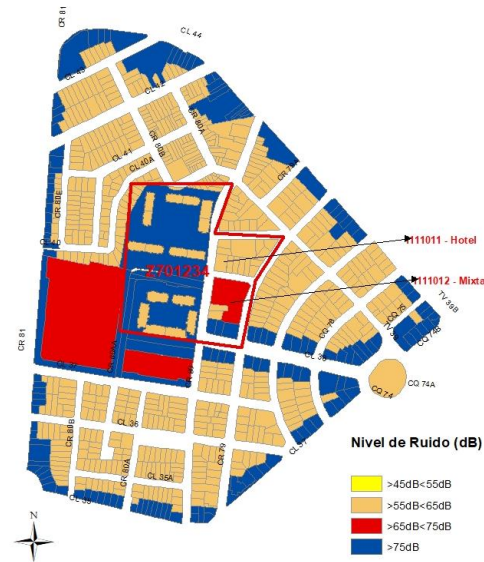
**Tabla 8.** Tabla de datos del concepto-clase *medición*.

MEDICIÓN							
Cód.	Fecha	Id Ingeniero	Cód Zona	Suma Valores	Cantidad Registros	Valor Promedio	Alerta Final
M4031004	5/03/2018	1893213	Z70234	379 dB	5	76 dB	Azul

## 6. Conclusiones y trabajo futuro

La definición de las buenas prácticas ayuda a identificar los elementos que se requieren en la abstracción de geodatos para representar eventos de ruido ambiental y generar mapas de ruido según zonas específicas de estudio en las fases de educación de requisitos y de planeación y diseño de proyectos SIG.

Para la representación de eventos de ruido ambiental se toman como base los esquemas preconceptuales para generar el catálogo de objetos geográficos, el modelo



**Fig. 5.** Mapa de Ruido de la zona de estudio.

de geoalmacenamiento y el catálogo de representación de objetos geográficos según las buenas prácticas de educación geoespacial de requisitos en los proyectos SIG para generar mapas de ruido.

La representación propuesta ayuda a la comprensión del proceso de medición de eventos relacionados con ruido ambiental, ya que presenta los objetos geográficos, los procesos y eventos que interactúan en el proceso.

Como trabajo futuro se pueden representar eventos naturales a partir de modelos matemáticos que ayuden a simular los eventos desde dominios de software científico para prevenir riesgos y desastres.

**Agradecimientos.** Este artículo es producto del proyecto de investigación Doctoral: *Una extensión a esquemas preconceptuales para el refinamiento en la representación de eventos y la notación matemática*, con código Hermes 39886, de la Universidad Nacional de Colombia, que financia Colciencias en la convocatoria 727 de becas para estudiantes de doctorado en Colombia. Además, a la Universidad de San Buenaventura, Medellín, por apoyar el proyecto de investigación Doctoral: *Definición de buenas prácticas de desarrollo de Sistemas de Información Geográfica utilizando el núcleo de Semat*, con el proyecto de investigación “Competencias Semat para un equipo de desarrollo de Proyectos SIG”, con código M4832.

## Referencias

1. Bejarano, J. S.: Gestión del ruido ambiental en Valencia. *Modelling in Science Education and Learning*, 11(1), pp. 25–42 (2018)

2. Zapata, C.M.: The UNC-Method revisited: elements of the new approach. Saarbrücken: Lambert (2012)
3. Zapata, C.M., Noreña, P.A., Vargas, F.A.: The Event Interaction Game: Understanding Events in the Software Development Context. *Development in Business Simulation and Experiential Learning*, 41, pp. 256–263 (2014)
4. Noreña, P.A., Zapata, C.M.: A Game for Learning Event-Driven Architecture: Pre-conceptual-Schema-based Pedagogical Strategy. *Development in Business Simulation and Experiential Learning*, 45, pp. 312–319 (2018)
5. OMG, Object Management Group: Business Process Model and Notation BPMN. Standard Document. <http://www.omg.org/spec/BPMN/1.2> (2016)
6. Murphy, E., King, E.A.: Smartphone-based noise mapping: integrating sound level meter app data into the strategic noise mapping process. *Science of The Total Environment*, 562, pp. 852–859 (2016)
7. Wei, W., Van, T., De Coensel, B., Botteldooren, D.: Dynamic noise mapping: A map-based interpolation between noise measurements with high temporal resolution. *Applied Acoustics*, 101, pp. 127–140 (2016)
8. Li, N., Feng, T., Wu, R.: Flexible distributed heterogeneous computing in traffic noise mapping, in *Computers. Environment and Urban Systems*, 65, pp. 1–14 (2017)
9. Herman, L., Řezník, T.: Web 3D visualization of noise mapping for extended INSPIRE buildings model. In: *International Symposium on Environmental Software Systems, Neusiedl am See* (2013)
10. Rao, A., Han, W.: An Adaptive Gaussian Particle Filter based Simultaneous Localization and Mapping with dynamic process model noise bias compensation. In: *IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS), IEEE Conference on Robotics, Automation and Mechatronics (RAM), Siem Reap, Cambodia* (2015)
11. Socoró, J.C., Ribera, G., Sevillano, X., Alías, F.: Development of an Anomalous Noise Event Detection Algorithm for dynamic road traffic noise mapping. In: *Proceedings of the 22nd International Congress on Sound and Vibration (ICSV22)* (2015)
12. Cai, M., Zou, J., Xie, J., Ma, X.: Road traffic noise mapping in Guangzhou using GIS and GPS, in *Applied Acoustics*, 87, pp. 94–102 (2015)
13. Gómez, D.: Resolución espacial en la elaboración de mapas de ruido por interpolación. *Ingenierías USBMed*, 8(1), pp. 56–62 (2017)
14. Suárez, E., Barros, J. L.: Traffic noise mapping of the city of Santiago de Chile. *Science of the total environment*, 466, pp. 539–546 (2014)
15. Bozkurt, T.S., Demirkale, S. Y.: The field study and numerical simulation of industrial noise mapping. *Journal of Building Engineering*, 9, pp. 60–75 (2017)
16. Zhao, J., Qin, Q., Xie, C., Wang, J., Meng, Q.: An efficient method of predicting traffic noise using GIS. In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (2013)
17. Bajarano, J., Diago, S.: Gestión del ruido ambiental en Valencia. *Modelling in Science Education and Learning*, 11(1), pp. 25–42 (2018)
18. Cohen, A.M., Salinas, O.: Contaminación auditiva y ciudad caminable. In: *Estudios demográficos y urbanos*, 32(94), pp. 65–96 (2017)
19. Durango, C.: Asociación de datos espacio-temporales en bases de datos Oracle. *Ingenierías USBMed*, 5(2), pp. 100–108 (2014)



20. Torres, D.M., Zapata, C.M.: Best practices of interoperability among heterogeneous software systems: a Semat-based representation. *Revista Facultad de Ingeniería*, 26(44), pp. 155 (2017)
21. Durango C.E., Zapata, C.M.: Una representación basada en Semat y RUP para el Método de Desarrollo SIG del Instituto Geográfico Agustín Codazzi. *Ingenierías USBMed*, 6(1), pp. 24–37 (2015)
22. Casas, O., Betancur, C.M., Montaña, J.S.: Revisión de la normatividad para el ruido acústico en Colombia y su aplicación. *Entramado*, 11(1), pp. 264–286 (2015)



## Diseño de un sistema de suministro de energía para vehículos eléctricos usando lógica difusa

Ismael Osuna Galán<sup>1</sup>, Yolanda Pérez Pimentel<sup>1</sup>, Juan Villegas Cortez<sup>2</sup>,  
Carlos Avilés Cruz<sup>2</sup>

<sup>1</sup> Universidad Politécnica de Chiapas,  
México

<sup>2</sup> Universidad Autónoma Metropolitana Azcapotzalco,  
México

{iosuna,ypimentel}@upchiapas.edu.mx, {juanvc, caviles}@correo.azc.uam.mx

**Resumen.** La tecnología de los sistemas electrónicos de potencia se ha diversificado en áreas industriales, comerciales y residenciales. Actualmente, los carros eléctricos usan diversos controladores que les permiten tener una eficiencia comparable con los carros convencionales. Los vehículos eléctricos (EV) son sistemas mecatrónicos complejos descritos por modelos no lineales y, por lo tanto, su diseño y análisis de control no es una tarea fácil. De igual forma, incrementar el rendimiento de un EV no depende de crear mejores sistemas de almacenamiento. Este artículo muestra el desarrollo de un sistema de gestión energética basado en lógica difusa para un vehículo eléctrico con la finalidad de minimizar su consumo total de energía y optimizar el banco de baterías. Los resultados experimentales usando el controlador difuso se comparan con los resultados en condiciones normales de operación. Se observa un incremento en el rendimiento de las baterías y del rendimiento en general del consumo de energía. Las señales de velocidad adquiridas muestran una mejora en algunos parámetros dinámicos, tales como el sobreimpulso, el tiempo de establecimiento y el error de estado estable. Se muestra que este controlador difuso aumenta la eficiencia energética general del vehículo.

**Palabras clave:** lógica difusa, vehículos eléctricos, control de energía.

## Design of a Power Supply System for Electric Vehicles using Fuzzy Logic

**Abstract.** The technology of electronic power systems has diversified into industrial, commercial and residential areas. Currently, electric cars use various controllers that allow them to have an efficiency comparable to conventional cars. Electric vehicles (EV) are complex mechatronic systems described by non-linear models and, therefore, their design and control analysis is not an easy task. Similarly, increasing the performance of an EV does not depend on creating better storage systems. This article shows the development of an energy management system based on fuzzy logic for an electric vehicle in order to minimize its total energy consumption and optimize the battery bank. The

experimental results using the fuzzy controller are compared with the results under normal operating conditions. There is an increase in the performance of the batteries and the overall performance of the energy consumption. Acquired velocity signals show an improvement in some dynamic parameters, such as overshoot, set-up time and steady-state error. It is shown that this fuzzy controller increases the overall energy efficiency of the vehicle.

**Keywords:** fuzzy logic, electric vehicles, energy control.

## **1. Introducción**

Con las regulaciones de emisiones contaminantes, los avances en la tecnología de motores eléctricos y la creación de baterías de alto desempeño, los fabricantes de automóviles alrededor del mundo han comenzado a considerar más seriamente la comercialización de vehículos que usen fuentes de energía alternativas.

La tecnología aplicada a los vehículos eléctricos (EV) ha logrado un desempeño comparable con los motores de combustión interna, dado que los motores de tracción eléctrica proporcionan una aceleración rápida y el motor de combustión interna funciona bien a velocidades constantes es que se han creado los vehículos híbridos [6, 7].

Hoy en día, la gran mayoría de los vehículos eléctricos en el mercado usan una única fuente de energía basada en baterías para generar el funcionamiento del vehículo. La baja densidad de potencia y el corto ciclo de vida es una de las deficiencias de la batería electroquímica. La aparición de sistemas de almacenamiento de energía compuesto tales como celdas solares, baterías de alto rendimiento, sistemas electrónicos de potencia son grandes avances.

El uso de tecnología de electrónica de potencia junto con un sistema que controle el consumo de energía del banco de baterías reduce el ciclo de descarga de corriente de las baterías y aumenta la distancia de desplazamiento del EV. [8]

Este artículo propone un enfoque para la administración de energía en aplicaciones en el EV UPChis01 (Fig. 1) basado en la reducción de la corriente de potencia aplicada al banco de baterías. Una de las principales ventajas de esta propuesta es la introducción de un sistema difuso capaz de mejorar la vida útil del banco de baterías y mejorar su rendimiento [2,5]. Este artículo está organizado de la siguiente forma:

Primero se muestra las especificaciones y características del vehículo eléctrico y el modelo que es usado para este estudio. Después se describirán el Sistema difuso usado para la gestión de potencia. Se muestran los resultados experimentales de la solución propuesta y analiza la limitación dinámica de la corriente de acuerdo con el estado de carga del banco de baterías. Finalmente, las conclusiones se harán en la última sección.

## **2. Trabajos relacionados**

En México, diversas universidades han desarrollado EV's que cumplen estándares internacionales, la Facultad de Ingeniería de la UNAM recientemente construyó el auto "Kalani" el cual es un vehículo monoplaza tripoide, con una estructura de acero y fibra de vidrio. Tiene un peso de 50 kilos y mide 120 centímetros de ancho por 220



**Fig. 1.** Carro eléctrico UPChis01.

centímetros de largo y 80 centímetros de altura. Cuenta también con celdas de litio de 1000 watts y una eficiencia de 14 kilómetros de distancia en una sola carga.

Por otra parte, la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME) del Instituto Politécnico Nacional (IPN) modificaron la estructura y diseño de un auto Volkswagen sedán para construir un carro con un motor de corriente directa que funciona con un rango de voltaje de 36 a 92 voltios, que es alimentado con un banco de seis baterías de ácido plomo de ciclo profundo de 8 V cada una. El Centro de Investigación en Mecatrónica Automotriz del Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), campus Toluca, actualmente se encuentran desarrollando un vehículo eléctrico para ser comercializado por empresas para la distribución de sus productos dentro de ciudades, este proyecto está en coordinación con la Secretaría de Energía.

Para garantizar la utilización completa de la energía disponible de las baterías de un EV a diferentes velocidades, la mayoría de los esquemas utilizan un controlador PID para administrar la energía de las baterías. El controlador PID convencional requiere un poco de ajuste para obtener una respuesta rápida y dinámicamente aceptable. De nuevo, generalmente se implementa utilizando circuitos amplificadores operacionales cuyos parámetros se ajustan para un punto de operación basado en un modelo lineal por piezas del sistema no lineal. Estos circuitos se ven afectados por el tiempo de uso y la temperatura, lo que causa la degradación del rendimiento del sistema.

En diversos artículos [2,5,10] se presentan controladores basados en sistemas de lógica difusa para controlar la potencia de salida de un inversor de ancho de pulso (PWM) utilizado en un esquema de conversión de energía de un motor eléctrico. El generador de inducción autoexcitado utilizado en esos motores tienen el problema inherente de fluctuaciones en la magnitud y frecuencia de su voltaje con los cambios en la velocidad. Para evitar ese inconveniente, la magnitud variable, el voltaje de frecuencia variable en los terminales del generador se rectifica y la potencia de CC se transfiere a la carga a través de un inversor PWM. El objetivo es rastrear y extraer la máxima potencia del sistema de energía y transferir esta potencia a la carga aislada local.

### 3. Modelado del carro eléctrico

#### 3.1. Componentes del carro eléctrico

Primeramente, se describe características generales y componentes del vehículo eléctrico. Vehículo VW originalmente de motor de combustión interna de 4 cilindros, modelo del año 2000 con una masa de 550 kg. La descripción básica de los componentes del EV UPCis01 está descrito en la tabla 1.

Existen diferentes tipos de arquitectura [6] algunas posibilidades son: 1 a 4 máquinas eléctricas, Maquinas eléctricas AC o DC, con o sin caja de cambios, alto o bajo voltaje en baterías, una o 3 fases de carga. La arquitectura elegida es la mostrada en la Fig. 2.

#### 3.2. Análisis matemático

Las fuerzas que el vehículo eléctrico debe superar son fuerzas debido a la gravedad, viento, resistencia a la rodadura y efecto inercial [3]. Dichas fuerzas se pueden observar en la Fig. 3.

La fuerza de tracción de un vehículo puede ser descrito por las dos ecuaciones siguientes:

$$F_{traccion} = \underbrace{M_{EV} \dot{v}_{EV}}_{F_{inercia}} + \underbrace{M_{EV} g \sin(\alpha)}_W + \text{signo}(v_{EV}) \underbrace{M_{EV} g \cos(\alpha) c_{rr}}_{F_{friccion}} + \underbrace{\text{signo}(v_{EV} + v_{viento}) \frac{1}{2} \rho_{aire} C_{arrastre} A (v_{EV} + v_{viento})^2}_{F_{viento}},$$

$$C_{arrastre} = 0.01(1 + \frac{3.6}{100} v_{EV}).$$

Las descripciones de cada una de las variables en el modelo se muestran en la tabla 2.

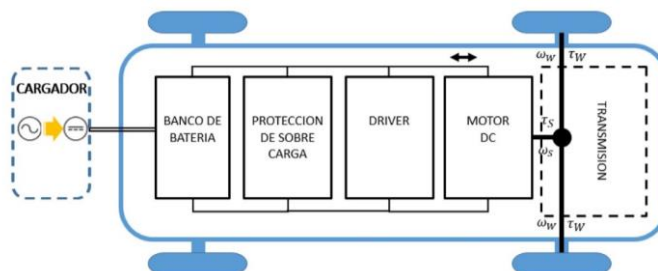
#### 3.3. Baterías eléctricas

Estos son rangos generales de voltaje para baterías de 6 celdas de plomo y ácido:

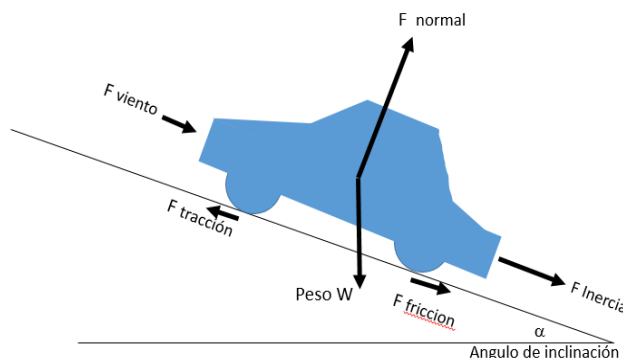
1. Circuito abierto (inactivo) a plena carga: 12,6 V ~ 12,8 V (2,10 ~ 2,13 V por celda).
2. Circuito abierto a plena descarga: 11,8 V ~ 12,0 V.
3. Cargado a plena descarga: 10,5 V.

**Tabla 1.** Descripción de los componentes del EV.

Cantidad	Descripción
1	Motor eléctrico DC
1	Controlador de motor
12	Baterías 6 VCD
1	Transmisión manual 5 vel. y 1 reversa
4	Llantas rin 16



**Fig. 2.** Principales componentes del EV.



**Fig. 3.** Análisis de fuerzas del EV.

4. Carga continua de preservación (flotación): 13,4 V para electrolito de gel; 13,5 V para AGM (absorbed glass mat), y 13,8 V para celdas de electrolito fluido común.
5. Todos los voltajes están referenciados a 20°C, y deben ajustarse -0,022 V/°C por cambios en la temperatura.
6. Las recomendaciones sobre el voltaje de flotación varían, de acuerdo con las recomendaciones del fabricante.
7. Una tensión de flotación precisa ( $\pm 0,05$  V) es crítica respecto a la longevidad; muy baja (sulfatación) es casi tan mala como muy alta (corrosión y pérdida de electrolito)

**Tabla 2.** Descripción de parámetros del modelo del EV.

PARÁMETRO	DESCRIPCIÓN	UNIDADES
$F_{traccion}$	Fuerza de tracción	N
$F_{inercia}$	Fuerza de inercia	N
$F_{friccion}$	Fuerza de fricción	N
$W$	Peso de EV	N
$F_{normal}$	Fuerza normal	N
$F_{viento}$	Fuerza de resistencia debida al viento	N
$\alpha$	Angulo de la superficie	Radianes
$M_{EV}$	Masa de EV	Kg
$v_{EV}$	Velocidad de EV	$\frac{m}{s}$
$\dot{v}_{EV}$	Aceleración de EV	$\frac{m}{s^2}$
$\rho_{aire}$	Densidad del aire a 30°C 1.650	$\frac{kg}{m^3}$
$A$	Área frontal	$m^2$
$c_{rr}$	Resistencia del neumático al rodamiento	
$C_{arrastre}$	Coefficiente de fricción aerodinámico	
$v_{aire}$	Velocidad del viento	$\frac{m}{s}$

8. Carga típica (diaria): 14,2 V a 14,5 V (dependiendo de las recomendaciones del fabricante).
9. Carga de ecualización (baterías de electrolito fluido): 15 V para no más de 2 horas. La temperatura de la batería debe controlarse.
10. Después de plena carga la tensión de terminales caerá rápidamente a 13,2 V y luego lentamente a 12,6 V.

#### 4. Construcción del sistema difuso

En esta sección se describe el desarrollo del sistema difuso. En [4] se creó una prueba experimental para obtener la energía eléctrica consumida por un coche eléctrico de juguete, utilizada como prueba experimental para los controladores propuestos. El objetivo principal de este trabajo fue aplicar un controlador de lógica difusa para verificar si el controlador mejora el consumo de energía del EV. Siguiendo esas ideas,



se desarrolló un sistema de inferencia difuso tipo Mamdani. Se usó el software LabVIEW para implementar ese sistema de forma embebida en un controlador CompactRIO de la empresa National Instruments.

Es un controlador tipo MISO (entradas múltiples y salida única), que tiene como entradas: pendiente de camino, profundidad de descarga y velocidad de conducción; como salida: la nueva velocidad del vehículo.

#### **4.2. Conjuntos difusos**

En este caso el sistema difuso es un administrador de división de potencia y controla todos los sistemas incorporados en el vehículo. Mediante una red de controlador CAN (acrónimo del inglés Controller Area Network) que es un protocolo de comunicaciones basado en una topología bus para la transmisión de mensajes en entornos distribuidos, dicha red es utilizada para comunicar todos los dispositivos.

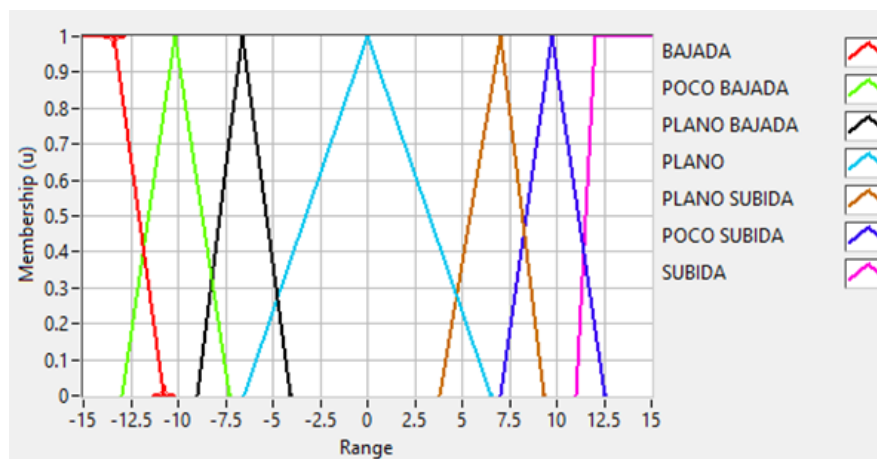
Se muestran las variables de entrada y salida programadas así como las variables lingüísticas. Algunas razones para describir por qué usar funciones de pertenencia trapezoidal y triangular se mencionan en [9], donde se hace una comparación entre distintas funciones de pertenencia, señalando las mejores situaciones para cada uso. Los tres casos especiales para la opción Trapezoidal se destacan a continuación:

1. **Construcción:** se refiere a los métodos para obtener las funciones de membresía. Generalmente, hay dos métodos para crear una función de pertenencia: Model-Driven se necesita un modelo matemático para describir la planta y utilizan técnicas de optimización para ajustar los parámetros. Es más fácil de modelar variables si se hace una linealización como es el caso para controlar el combustible requerido por el controlador de potencia.
2. **Monotonidad:** Describe cómo un sistema conserva la estructura original durante un proceso. Una función monótona se expresa como una expresión matemática que no cambia el orden dado. Una función trapezoidal funciona mejor que Gaussiana y otros;
3. **Costo computacional:** en tareas de control en tiempo real, se prefiere un algoritmo de bajo costo, en otras palabras, el controlador lleva a cabo un proceso más rápido. Al hacer un sistema de Mamdani con 50 reglas o menos que eso una función del tipo Gaussiano es mejor. Sin embargo, con el caso de este estudio, se utilizan más de 100 reglas, una inferencia con trapecios y triángulos se realiza más rápido.

#### **4.3. Reglas difusas**

Las tareas del controlador de disminuir la velocidad EV para reducir el consumo de energía del vehículo. En primer lugar, la pendiente de las entradas, derivadas del gradiente de la carretera, profundidad de descarga, calculada a partir del modelado del EV y la velocidad del vehículo.

Se usaron 224 reglas, el controlador difuso actúa en situaciones donde la profundidad de descarga es mayor al 70% y se aplica directamente en el rendimiento del vehículo, el sistema disminuye la velocidad para proteger la energía de la batería al reducir en el



**Fig. 4.** Descripción de la variable de entrada PENDIENTE.

consumo de energía eléctrica, ya que es directamente proporcional a la fuerza de tracción del EV. Este controlador funciona en diferentes entornos (pendiente llana de la carretera, escenarios ascendentes y descendentes).

Por ejemplo, una de estas reglas es la siguiente:

"PENDIENTE = PLANO and DESCARGA = ECO MODE y VELOCIDAD = ALTA then VELOCIDAD CORREGIDA = MEDIA BAJA".

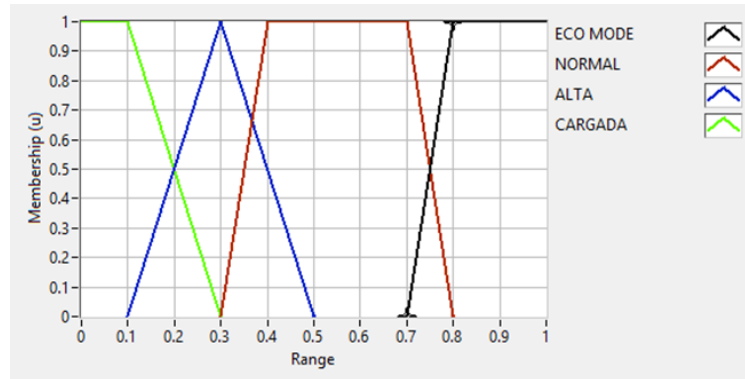
En este caso, el controlador difuso reduce la velocidad (VELOCIDAD CORREGIDA) al detectar una descarga de las baterías superior al 70% (conjunto ECO MODE) por lo que la velocidad solicitada por el usuario es ignorada por las condiciones de carretera (conjunto PLANO).

## 5. Pruebas experimentales

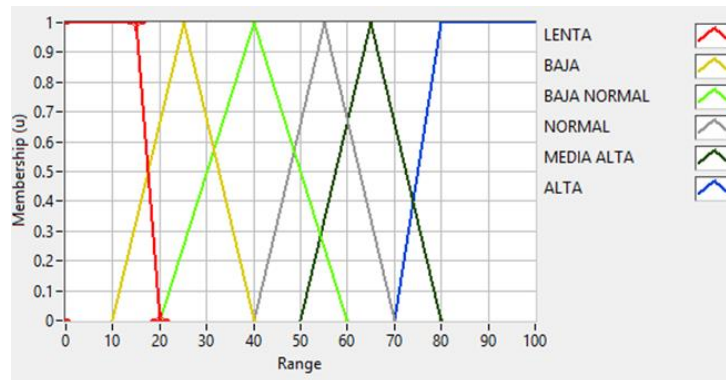
Un ciclo de conducción se compone de micro-viajes y tiene un período de 5 a 40 minutos. Esta duración debe contener suficientes micro-viajes que reflejen el comportamiento de conducción en el mundo real.

Los ciclos de manejo pueden ser de laboratorio o reales [4]. Los siguientes son algunos parámetros básicos del ciclo fue realizado por un tramo de la nueva carretera Tuxtla Gutiérrez – Suchiapa-Villaflora; Duración: 499 s; Distancia recorrida: 5.4 km; Velocidad media: 60.2 km / h; Velocidad máxima: 85.5 km / h. La velocidad del ciclo de conducción y la velocidad controlada se expresan en la Fig. 8.

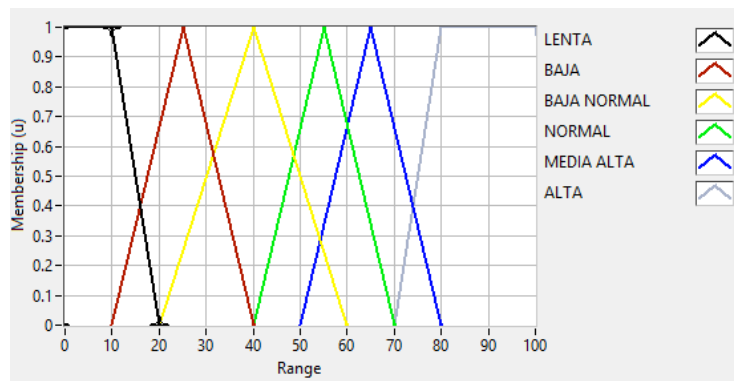
En cualquier caso, es importante obtener variables principales como velocidad, aceleración, distancia y pendiente de la ruta. Cuando se habla sobre el desarrollo del ciclo de manejo, tres pasos son importantes: selección de ruta, recolección de datos y construcción de ciclo. La selección de ruta implica seleccionar el curso para describir el ciclo. Consiste en determinar si la ruta es una autopista con velocidad constante, vías arteriales o conducción urbana, por ejemplo.



**Fig. 5.** Descripción de la variable de entrada DESCARGA.



**Fig. 6.** Descripción de la variable de entrada VELOCIDAD.



**Fig. 7.** Descripción de la variable de salida VELOCIDAD CORREGIDA.

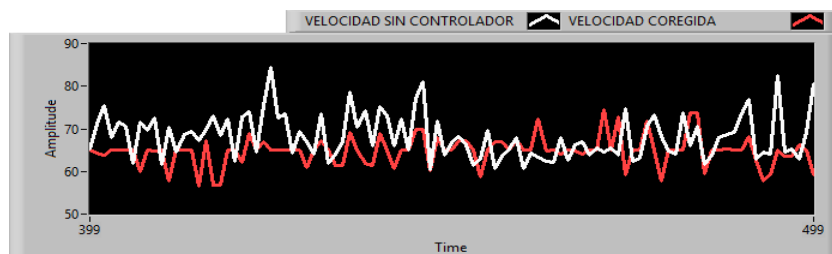
La recopilación de datos es la capacidad de recopilar los parámetros, los datos con los sensores adecuados para describir el ciclo de conducción. Finalmente, para la

**Tabla 3.** Valores promedio de pérdida de potencia durante un ciclo de viaje.

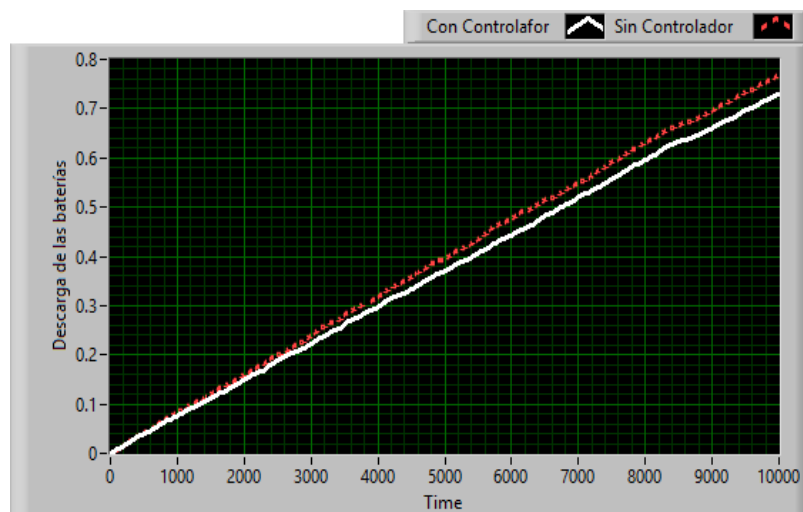
PARÁMETRO	RESULTADO EV	MEJORA
Pérdida de potencia sin controlador	1479 W	
Pérdida de potencia con controlador	1135 W	23.25 %

**Tabla 4.** Valores promedio de corriente durante un ciclo de viaje.

PARÁMETRO	RESULTADO EV	MEJORA
Banco de baterías sin controlador	34.95 A	
Banco de baterías con controlador	26.44 A	24.34 %



**Fig. 8.** Velocidades del EV durante un mini viaje.



**Fig. 9.** Tiempo de descarga del banco de baterías del EV.

construcción del ciclo se dividen todos los datos en micro viajes y se construye una función de dominio del tiempo de la velocidad del vehículo.

En las siguientes tablas, se muestran las pérdidas de potencia del motor eléctrico. Con esas pérdidas, se produjo la pérdida de calor en la batería. Los valores medios obtenidos en los resultados experimentales muestran una mejora general del 23.7% con la aplicación del sistema difuso. La tabla 3 y tabla 4 dan muestra de la mejora en el rendimiento del banco de baterías.

En la Fig. 9, se ilustra la profundidad de la descarga en función del tiempo. La mayor autonomía del vehículo es visible cuando el ciclo con el controlador tiene un intervalo de tiempo más largo para drenar toda la batería.

El ciclo de conducción se repite hasta que ciclo de descarga alcanza el 95%, que es un estado de batería vacía para analizar la distancia adicional dada por el sistema desarrollado.

## **6. Conclusiones**

Las pruebas experimentales muestran que con el controlador integrado en el EV se obtuvieron los resultados deseados. En todos ellos, se generaron valores más bajos para todos los parámetros analizados, lo que demuestra la efectividad de la implementación del sistema. El consumo de energía eléctrica tuvo una reducción en todas las pruebas, dando integridad y seguridad al banco de baterías [1].

El tiempo más largo alcanzado por el vehículo es un resultado necesario, evitando la batería vacía y dando la oportunidad al conductor de encontrar una próxima estación de recarga otorgando una pequeña ganancia en la distancia de viaje.

Será necesario verificar si el comportamiento para la autopista tiene el mismo comportamiento el sistema que en un medio urbano.

Es probable que la velocidad se vea reducida de manera natural por las condiciones de tráfico y límites de velocidad que ocurren en una ciudad.

Es de suponer que en la mayoría de los escenarios de autopista ocurran mayores velocidades por lo que el EV puede ser una solución viable para la comunicación entre pequeñas poblaciones cercanas a la ciudad. Por lo tanto, para mejorar el análisis del sistema y realizar diferentes investigaciones sobre los temas adoptados en el proyecto.

Las sugerencias para futuros estudios son en el sentido de mejorar el modelo del motor eléctrico usando un modelo de batería Li-Ion o la aplicación nuevos tipos de controladores como PID difuso o redes neuronales.

## **Referencias**

1. García-López, M., et al.: Análisis de pérdidas en semiconductores de potencia generadas por controladores difusos de velocidad en motores de CD sin escobillas. *Research in Computing Science*, 135, pp. 115–127 (2017)
2. Hanane, H., Ghouili, J., Cheriti, A.: A real time fuzzy logic power management strategy for a fuel cell vehicle. *Energy conversion and Management* 80, pp. 63–70 (2014)
3. Hibbeler, R.: *Engineering Mechanics Dynamics*. [S.l.]: Upper Saddle River, NJ: Pearson Prentice Hall (2016)
4. Howey, D.A., et al.: Comparative measurements of the energy consumption of 51 electric, hybrid and internal combustion engine vehicles. *Transportation Research Part D: Transport and Environment*, 16, pp. 459–464 (2011)

5. Kandi, M., Soleymani, M., Ghadimi, A.A.: Designing an optimal fuzzy controller for a fuel cell vehicle considering driving patterns. *Scientia Iranica. Transaction B, Mechanical Engineering*, 23, pp. 218–235 (2016)
6. Larminie, J., Lowry, J.: *Electric vehicle technology explained*. John Wiley & Sons (2012)
7. Maia, R., et al.: Electric vehicle simulator for energy consumption studies in electric mobility systems. In: *Integrated and Sustainable Transportation System (FISTS)*, 2011 IEEE Forum on. IEEE (2011)
8. Tariq, M., Kolhe, M., Doyle, A.: *Electric Vehicles: Prospects and Challenges*. Elsevier (2017)
9. Dongrui, W.: Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers. In: *International Conference on Fuzzy Systems (FUZZ-IEEE)* (2012)
10. Al-Jazaeri, A.O., Samaranayake, L., Longo, S., Auger, D.J.: Fuzzy logic control for energy saving in autonomous electric vehicles. In: *The IEEE International Electric Vehicle Conference (IEVC)*, pp. 1–6 (2014)

Impreso en los Talleres Gráficos  
de la Dirección de Publicaciones  
del Instituto Politécnico Nacional  
Tresguerras 27, Centro Histórico, México, D.F.  
junio de 2018  
Printing 500 / Edición 500 ejemplares

