Advances in Intelligent Learning Environments

Research in Computing Science

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 47,** septiembre 2012. Imagen de la portada: miriadna.com. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No.: 04-2005-121511550100-102, 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: *Juan Humberto Sossa Azuela, RFC SOAJ560723*

Research in Computing Science *is published by the Center for Computing Research of IPN.* ***Volume 47****, September 2012. 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.*

Volume 47

Volumen 47

Advances in Intelligent Learning Environments

Volume editors:
Editores del volumen

Ramón Zatarain Cabada
María Lucía Barrón Estrada
Yasmín Hernández Pérez

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2012

ISSN: 1870-4069

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface
(Prefacio)

This volume presents the reader a selection of papers on the theory and applications of intelligent learning environment.

Human beings are distinguished from each other mainly by the skills they possess to understand ideas or solve problems. This ability is closely linked to what we call intelligence. Artificial Intelligence (AI), has made that a computer can simulate an "intelligent" behavior, coming to solve problems that classical computing could not solve. The achievements in this area can be applied in almost any area of knowledge, including education, which has helped the development of tools such as Intelligent Tutoring Systems, Pedagogical Agents, and Computer Games in educational environments.

Today, learning and adaptive systems incorporate not only cognitive factors but also emotional or affective states of students. That means, that in order to build a modern learning systems we would need to study different fields of knowledge as pedagogy, computer science, educational psychology, and artificial intelligence. The aim of this book is to present different research works in some of the most interesting fields of intelligent learning systems.

In addition to the section devoted to intelligent learning environments, the volume also includes a selection of regular papers.

Ramón Zatarain Cabada
María Lucía Barrón Estrada
Yasmín Hernández Pérez

September 2012

Table of Contents

(Índice)

Page/Pág.

Intelligent Learning Environments

Regular Papers

Intelligent Learning Environments

Computer Systems for Analysis of Nahuatl

Carmen C. Martínez-Gil[1], Alejandro Zempoalteca-Pérez[1], Venustiano Soancatl-Aguilar[2], María de Jesús Estudillo-Ayala[3], José Edgar Lara-Ramírez[3], and Sayde Alcántara-Santiago[4]

[1]Universidad de la Cañada, Carr. Teotitlán-San Antonio Nanahuatipan Km. 1.7 s/n, Paraje Titlacuatitla, Teotitlán de Flores Magón, Oax., C.P. 68540, Mexico

[2]Universidad del Istmo, Carr. Chihuitan Ixtepec s/n, Ixtepec, Oax., C.P. 70110, Mexico

[3]Escuela de Ciencias, Universidad Autónoma Benito Juárez de Oaxaca. Av. Universidad s/n, Ex-Hacienda de 5 Señores, Oax., C.P. 68120, Mexico

[4]NovaUniversitas, Carretera a Puerto Ángel Km. 34.5, Ocotlán de Morelos, Oax., C.P. 71513, Mexico

{cmartinez, alejandro}@unca.edu.mx, venus@bianni.unistmo.edu.mx, salcantara@jacinto.novauniversitas.edu.mx

Abstract. This article describes two computer systems that allow us to analyze words written in the Nahuatl language. The main goal is the diffusion and preservation of indigenous language with great historical, linguistic, literary and nationalistic relevance by developing language resources for Nahuatl. One system automatically gets prefixes or suffixes of words from a text written in Nahuatl. This system was developed because Nahuatl writing contains agglutination, i.e. prefixes and/or suffixes are added to the root of a word to give it specific meaning. The other system is a Nahuatl to Spanish translator and vice versa, which also shows semantic information related to the terms in Nahuatl. This information includes the root, or roots of words as well as it´s grammatical category, which can be: a noun, adjective, pronoun, preposition, conjunction, article, adverb, verb or interjection. The system currently contains 1,514 terms.

Keywords: Computer systems, Nahuatl language, language resources, semantic information.

1 Introduction

Nahuatl is an indigenous language which is currently spoken in countries such as Mexico, El Salvador, United States, Guatemala and Nicaragua.

Currently, Nahuatl is the indigenous language most widely spoken in the Mexican territory, with approximately a million and a half fluent individuals, as reported by National Institute of Statistics and Geography and Computer. This language is valuable because it has great historical, linguistic, literary and nationalistic

significance. The states in Mexico where Nahuatl is still spoken include: State of Mexico, Puebla, Guerrero, Hidalgo, Veracruz, Oaxaca, Durango, Morelos, Mexico City, Tlaxcala, San Luis Potosi, Michoacán, Jalisco, among others.

Nahuatl is one of the American languages most studied and documented [1], [4-7], [9-11], there are several documents written in Nahuatl from which we can extract important information valuable to present and future generations.

Our interest in developing language resources for Nahuatl is primarily based on the following:

- Due to ignorance and mismanagement of the language terms, we are losing much of our culture. It is therefore important to accurately recognize and extract the information contained in documents written in Nahuatl;
- Preserving a language with historical roots, the loss of the Nahuatl language would represent the loss of part of the Mexican essence and identity. To better understand the cultures that still speak this language, as well as to communicate with people who only speak Nahuatl.

Moreover, the research field of Natural Language Processing (NLP) is a sub-discipline of Computer Science and Linguistics [2], [3], which is responsible for producing computer systems that facilitate the communication between man and man or man-machine using natural language. The purpose of NLP is to study the problems of automatic generation of natural language understanding. Some relevant applications of NLP are:

- Automatic Translation
- Speech Recognition
- Voice synthesis
- Extraction of information
- Information Retrieval
- Automatic generation of summaries
- Handwriting recognition
- Text Mining
- Question Answering

To build these applications, the NLP is assisted by linguistic resources.

Linguistic resources [8] are a set of language data in computer readable form and are used in the construction, improvement and evaluation of natural language systems, although the term also includes software tools or systems aimed to separate, collect, manage and use other resources. In this paper we present two software tools that allow us to analyze words in Nahuatl.

2 Development of the Computer Systems

In order to perform analysis of the Nahuatl language we have developed two software tools: one for prefixes and suffixes of words in Nahuatl text and another for the translation of the Nahuatl-Spanish terms, wich also provide semantic information of

words in Nahuatl. Because the Nahuatl is an agglutinative language, that is, prefixes and/or suffixes are added to the root of a word giving them complex meanings, it was necessary to develop a system that allows us to analyze the words in their most basic form. On the other hand, in order to develop future specialized applications, such as a stemmer, we require more information related to the word, since the mere corresponding meaning in Spanish will not be enough.

The steps followed to develop the system to obtain prefixes and suffixes are as follows:

1. System requirements. We searched and analyzed Nahuatl documents to form a collection of texts in txt format.
2. System design. The system design consists of three layer architecture: interface, application logic and storage.
3. Implementation and testing. The System was implemented in the programming language C #.

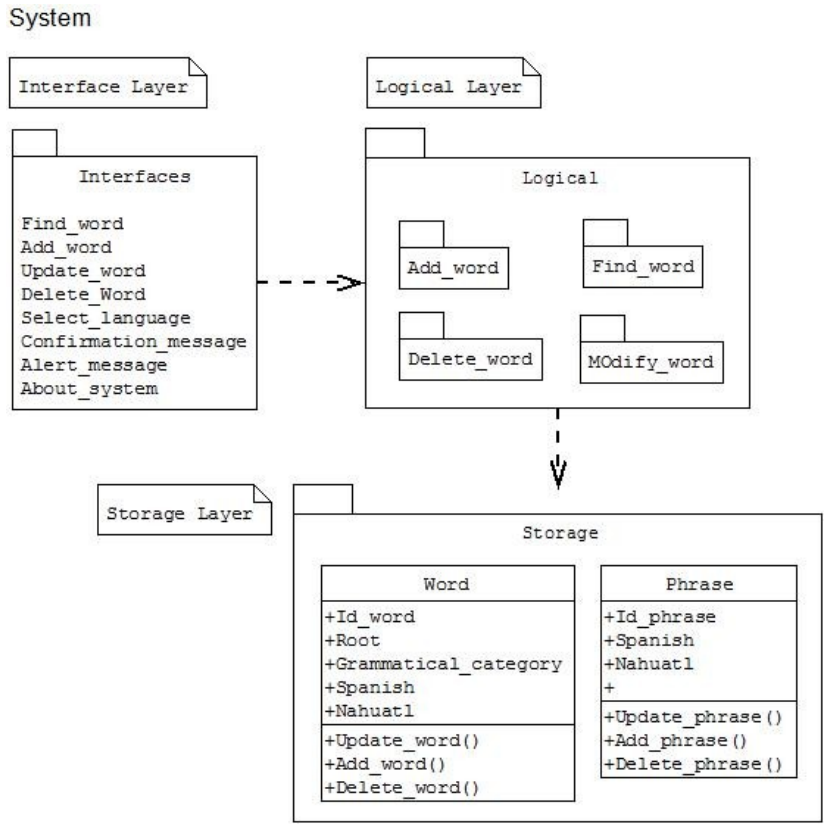

Fig. 1. Architecture of the translation and semantic information system.

Development of the Nahuatl-Spanish translator which includes semantic information was performed by the following process:

- System requirements. We identified the main functions of the system to keep the collection of terms, which are: search, add, modify and delete.
- System design. The system design consists of three layer architecture: interface, application logic and storage. Figure 1 shows the architecture interface of the system.
- Implementation and testing. The system was developed in C #, and the database was implemented in MS-Access 2000 because the database manager does not depend on a server to operate.

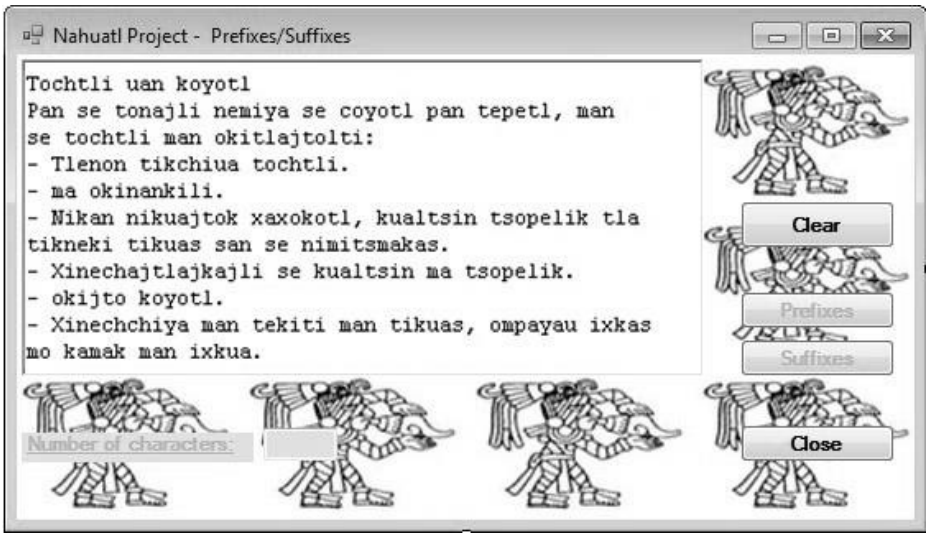

Fig. 2. Interface of the computer system to obtain prefixes and suffixes.

3 Results

Our results are two software systems, one for prefixes and suffixes of words in Nahuatl and another for the translation of the Nahuatl-Spanish terms and vice versa, which also shows semantic information related to the term in Nahuatl.

To test the first system we have a collection of 836 texts in Nahuatl classified into four categories: Poetry, Stories, Religion, and Miscellaneous. Figure 2 shows the interface of the system with an input text. Figure 3 shows the output of the system with prefixes of size 6 letters.

Figure 4 shows the computer system interface translation (using Spanish and Nahuatl languages) of terms and semantic information related to the term in Nahuatl. The semantic information is the root or roots of words and grammatical category,

which can be: a noun, adjective, pronoun, preposition, conjunction, article, adverb or verb. The system currently contains 1514 terms.

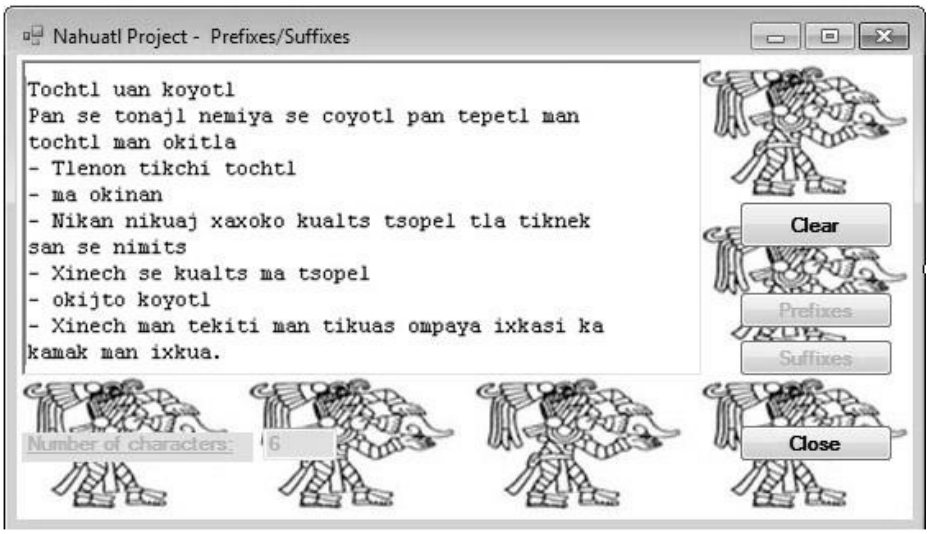

Fig. 3. Output of the computer system with prefixes of size 6.

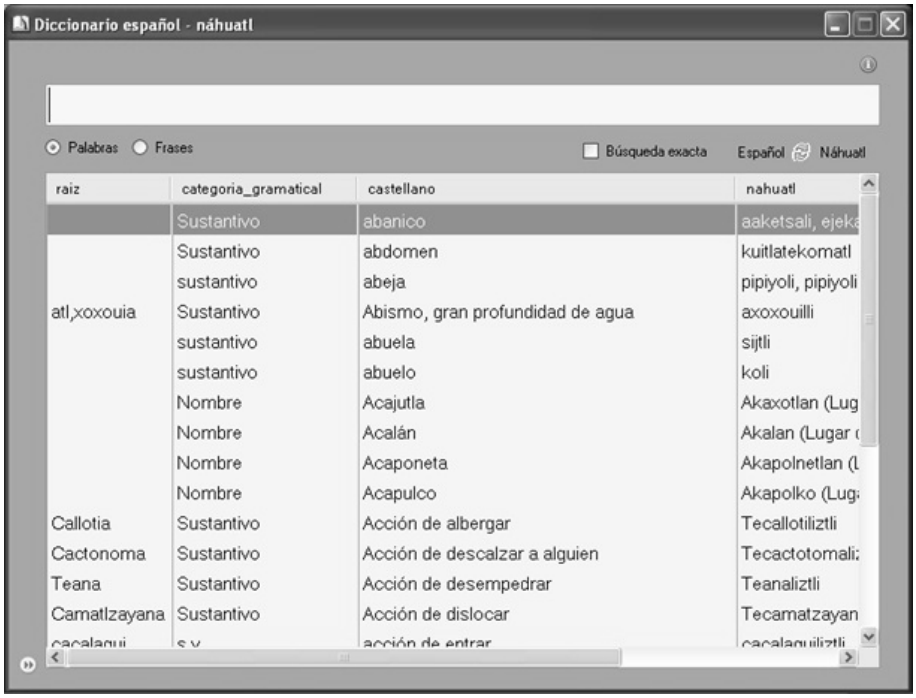

Fig. 4. Computer system interface for translation of terms and semantic information.

4 Conclusions

In an effort to preserve the use of pre-Hispanic Nahuatl language this article presents two computer systems that allow us to analyze words written in Nahuatl. One system automatically extracts the prefixes or suffixes of words from a text written in Nahuatl, and another system translates Nahuatl-Spanish terms and vice versa, and also shows semantic information related to the term in Nahuatl. Currently our computer system contains a database with 1,514 terms.

5 Future Work

As future work, we will develop a stemmer to continue the analysis and study of Nahuatl. To implement the stemmer we will use the two computer systems described in this work. As well as continue to develop linguistic resources for language Nahuatl such as part-of-speech tagging and statistical parsing.

References

1. Andrews, J. R.: Introduction to Classical Nahuatl. University of Oklahoma Press (2003)
2. Brill, E., Mooney, R. J.: An Overview of Empirical Natural Language Processing. AI Magazine. vol. 18, No. 4 (1997)
3. Bolshakou, I., Gelbukh A.: Computational Linguistics. Ciencia de la Computación. IPN-UNAM-FCE, México (2004)
4. Garibay K. Á. M.: Panorama Literario de los pueblos nahuas. Editorial Porrúa, México (1997)
5. Garibay K. Á. M.: La llave del náhuatl. 9ª edición. Editorial Porrúa. México (2007)
6. Langacker, R. W.: Studies in Uto-Aztecan Grammar. Moder Aztec Grammatical Sketches. Vol. 2. Summer Institute of Linguistics (1979)
7. Launey, M.: Introducción a la lengua y a la literatura Náhuatl. México D.F., UNAM. (1992)
8. Ortega-Mendoza R. M.: Descubrimiento Automático de Hipónimos a partir de Texto no Estructurado. Tesis de MAESTRÍA. Instituto Nacional de Astrofísica, Óptica y Electrónica (2007)
9. Saunders P., R.: A grammar of Tetelcingo (Morelos) Náhuatl. Journal of the Linguistic Society of America. Vol. 30, Num. 1 (1954)
10. Siméon, R.: Diccionario de la Lengua Náhuatl o Mexicana. [Paris 1885] Reprint: México (2001)
11. Wolgemuth, Carl: Gramática Náhuatl. Instituto Lingüístico de Verano. México (2002)

Objective Reviewer for Student Projects

Jesús Miguel García Gorrostieta[1], Jesús Pablo Lauterio Cruz[1], Indelfonso Rodríguez Espinoza[2], and José David Madrid Monteverde[1]

[1] Universidad de la Sierra, Moctezuma, Sonora, Mexico
[2] Universidad Tecnologica de Nogales, Nogales, Sonora Mexico

jesusmiguelgarcia@gmail.com, {jplauterio,indelfonso_rodriguez, jdmm}@hotmail.com

Abstract. In this paper, we present a web-based system to provide student aid in structuring research projects, specifically in the drafting of the objective. We use MOODLE as platform to present course material and to evaluate the student objective, we establish a dictionary of verbs, articles and tools for structural analysis of the objective with the implementation of a finite-state machine. This information is presented to students to receive acustomized feedback of their objective with an example of a well structured objective from the objectives repository. Finally we carried out an experiment with students´ final projects and applied a satisfaction survey of the objective reviewer system.

Keywords: Web-based learning systems, natural language processing, course management systems, Moodle.

1 Introduction

The use of natural language processing technologies applied to the study of texts for information analysis is widely used, as presented in the article by Muñoz [1] which performs the extraction of information in the domain of notary texts. Also Rose [2] shows a framework for retrieving text documents through natural language processing; this approach is based on the application of different techniques and rules that explicitly encode linguistic knowledge. Documents are analyzed on different linguistic levels by linguistic tools which incorporate text annotations within each level [3].

This paper aims to create a computer tool to provide student aid in structuring a researchproject, specifically in the drafting of the objective. This tool provides a theoretical framework for the drafting of objectives, analysisof the objective after the student has written it, and provides feedback to improve the objective. The tool uses dictionaries which, combined with a transition matrix, provide feedback based on certain pre-established parameters. Finally, the student, after using the tool, has a more refined objective, which will help the faculty adviser.

The use of natural language when considering the formation of higher education students cannot be ignored. One of these stages of formation is related to the generation and application of knowledge through research, which is usually placed in the last semesters of the academic program. According to the institution, various mechanisms are adopted that allow students to enter in the field of research, either through business internships, professional practice or in the various forms of professional qualification, all presenting the possibility of doing a research project. However, the process of drafting the research projects is usually not an easy task for students. Therefore, the system described in this paper intends to assist the work of the teacher and to facilitate and guide students through this process, specifically in the objective setting. This part in particular is important because it is the objective which shows the expected end result, besides being the guide that directs and allows monitoring the investigation in order to maintain a course leading to the goal initially proposed [4].

The analysis of natural language requires a lexical and grammatical analysis as we can see in the work of Dominguez[5] which implements an application for grammatical analysis to the Spanish language for database queries. Firstly performing a lexical analysis to check the input sentence, identify words and proceed to tag them using a lexical dictionary; in that dictionary are stored all the words that userspredominantly use, then a grammatical analysis is performed using a finite state machine to determine whether a sentence is grammatically valid. In this article we propose the use of a tool that integrates the analysis of a research objective in natural language to the structure of a course management system Moodle; we establish a dictionary of verbs which analyzes the number of verbs in the objective, and also analyzes the number of words used, finally a basic grammatical analysis of the objective is performed using a finite state machine.This information is presented to students to receiveimmediate feedback ontheir objective. Finally we carried out an experiment with students´ final projects.

2 The model

The model consists in a Moodle course online, in which we present several resources to write objectives, these resources must be reviewed by the student. After that the student answers a test and if his score is higher than 70, the option access to the objective reviewer is enabled in order to begin the redaction.

The student writes his objective and requests the analysis, the objective reviewer performs the following process: a lexical analysis and labeling, a count of words, verbs, articles and tools found in the objective, the word count is compared with the maximum number of 43 words and minimum of 13 words found in the repository of written objectives which has 100 sample objectives, the verb count is compared with the maximum number of verbs found in objective repository 5, the minimum value for verbs is one, which should be in the infinitive form as indicated by the drafting guide for the university.

Simultaneously, basic grammatical analysis of the objective is performed using a finite state machine with 4 states: Figure 1 presents the model used to analyze the

objective, with the automaton we ensure that the first element of the objective is a verbor anarticle in state 0 and subsequently a tool is used to achieve the objective in state 2 to finish the sentence. In case of finding any mistakes in the objective, the objective reviewer suggests how to improve the redaction. Once the student corrected all errors an option is enabled to answer the satisfaction survey instrument.

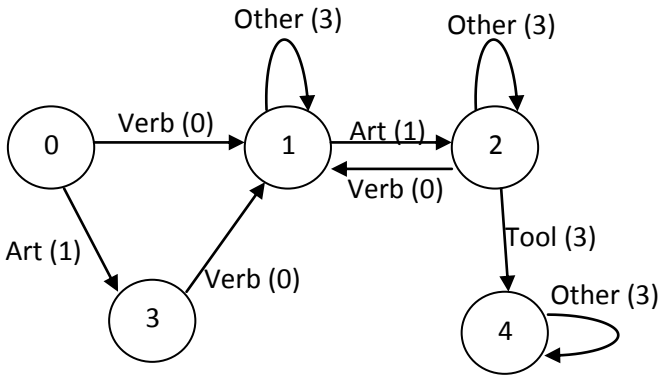

Fig. 1. Automaton used to analyze the objective.

The fourth state is the final one of state machine, where a verb and tool has been found successfully in the analyzed objective. When the process ends, the objective reviewer sends a success message to the student, if the fourth state in the automaton is not achieved, the objective reviewer indicatesa recommendation to improve the objective, in the absence of a verb the use of at least one verb in the infinitive form is recommended in order to define the action.In the case exceptions words are found, it isrecommended they are deleted, if the state machine is not successfully finish a message is send depending on the position which has failed.

For example, if the state in the automaton is 1 or 2, this indicates that it is necessary to include a tool in the objective, if thestate is 0 or 3, it indicates that it is necessary to start the objective with a verb in the infinitive form, finally, the objective reviewer shows an optimal objective which uses the same verb in the objective analyzed or a random objective.

3 Case Study

The experiment to test the tool was applied to a group of 42 students from three different Mexican Universities in the State of Sonora. Some of the students are doing research work whileothers are in advanced semesters in courses of research methodology. For this experiment we used the online course in Moodle. (http://moodle.moctezumavirtual.com). It starts by inviting all students to participate via email using the course “Intelligent Tutor for Research Projects” with access key “sonora”, here we indicate the importance of reading the material for writing adequate objectives. In Figure 2 we can see the content of the Moodle Course usedas first

material to teach students how to write an objective, and in the second block we have the test and the student satisfaction survey.

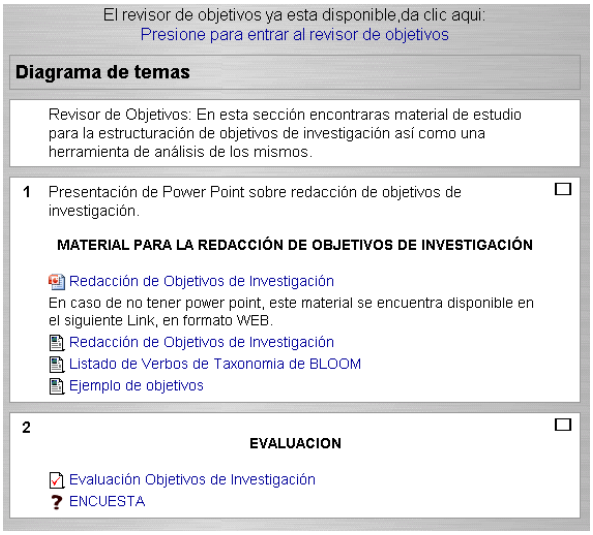

Fig. 2. Moodle Course "Intelligent Tutor for Research Projects".

After that, we ask students to answer a five question quiz to confirm they have read the material. When the student receives a positive score on the quiz, anoption is enabled touse the objective reviewer, and then the student can perform a preliminary analysis of his objective, before it is sent to the University's academic advisor.

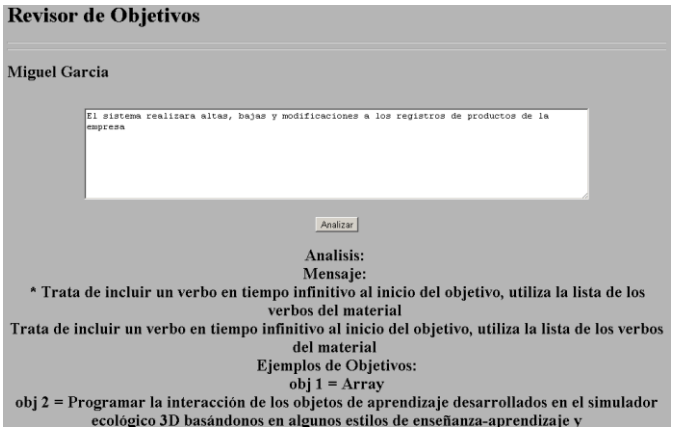

Fig. 3. Objective reviewer interface.

The interface of the objective reviewer is shown in Figure 3, where feedback to the students is shown. In this case the objective analyzed in spanish was: "El sistema

realizara altas, bajas y modificaciones a los registros de productos de la empresa" ("The system will perform additions, deletions and modifications to the records of the company's products"). The parser takes the first word "El" ("The") and identifies if it as an article. In the automaton this leads to state 3 in which a verb is expected but the second word is "sistema" ("system") category labeled "other" in this case the systems sends to user a feedback indicating that they must use a verb in infinitive form at the beginning of the objective. Once the students' achievement complies with all recommendations and they obtain a successful analysis, the student will answer a satisfaction survey to determine the objective reviewer utility.

4 Results

As a result of the 42 students who used the objective reviewer, there were 150 different types of feedback, the students made 186 attempts and the average usage time was 5:10 minutes. Figure 4 shows the numbers of attempts for each student to use the analyzer, the number of attempts were taken from the amount of feedbacks presented to the students.

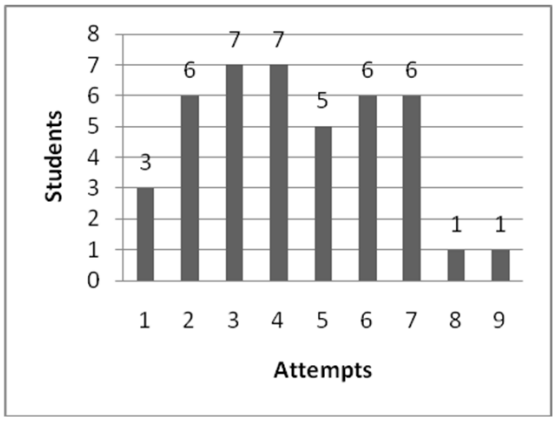

Fig. 4. Number of attempts per student.

By analyzing the types of messages sent to students it was found that 45% of the time the objective reviewer provided feedback to students such as include a tool, 22% were of the type include infinitive verb, 20% were suggestions to reduce the number of words in the objective and finally 13%" recommended they add words to the objective.

With this data we can concludethat most students did not include a tool in their objective, which refers to "how" the objective will be achieved, such a tool couldbe "web technology" or "Database."

The results of the satisfaction survey indicated 84% of the group rated the objective reviewer as of"great use", 13% of students indicated that it was moderately helpful and only 2% of students indicated that it was not useful.

We performed a statistical test to the results of objectives, hypothesis testing of a sample: applied under the t-student distribution, assuming that the data behaved in a continuous distribution.

Attempts were analyzed in the objective reviewer. One attempt meant that the student used it once and did not receive suggestions from the objective reviewer, 2 attempts or more, the student received feedback to improve its overall objective. We have the followings hypothesis:

Null hypothesis: The objective reviewer provides feedback to the student to formulate the overall objective.
Alternative hypothesis: The objective reviewer doesn't provide feedback to the student to formulate the overall objective.

The Interval found is (3.754, 5.045), where with 95% reliability, the average can take any value in the interval for our experiment we take 4.

Hypothesis Testing: $H0 = 4$ and $H1 \neq 4$.

We chose the statistical test "t-student" because of the small amount of unknown data as the deviation of the population:

Data: $Xn=4.4$, $Sn=2.07$, $u=4$, $n=42$.

$$T= (Xn - u) / (Sn/\sqrt{n}) = 0.4 / 0.3194 \quad \therefore T = 1.2524 . \tag{1}$$

The rejection area is located above $t= -2.02$ and $t= 2.02$, so the result of $T= 1.2524$ falls in the area of non-rejection. So we can conclude that the null hypothesis is not rejected and the instrument helped students receive feedback to improve the wording of their overall objective.

5 Conclusions and Future Works

The use of the objective reviewer in developing research projects is very useful for students who are often inexperienced in the correct wording of objectives and regularly require the teacher's personal advice; the parser proposed in this paper helps guide the student in the correct wording to directly analyze the text, recommending specific actions to improve the objectives was analyzed.

Using the objective reviewer we serve a large number of students and developed their objectives for the final review by the teacher. From the results of the satisfaction survey we can see that the use of the objective reviewer was useful for 84% of students. In future works, we shall attemptto analyze objectives written in English and we will optimize the system to work with mobile devices.

The objective reviewer is available to try in the follow internet address http://moodle.moctezumavirtual.com to use the system just sign in and enter to the course "Intelligent Tutor for Research Projects" with access key "sonora".

References

1. Muñoz, R., Montoya, A., Llopis, F., & Suarez, A.: Recognition of entities in the system EXIT. Natural Language Processing, vol. 23 (1998)
2. Rose G., C., & Olmos D., K.: Structured representation and retrieval of law enforcement documents using natural language processing and Extended Markup Language. In: Proc. of International Congress of Electronic Engineering, Vol. XXXIX, Chihuahua, México (2008)
3. Atserias, J., Carmona, J., Castellon, I., & Cervell, M.: Morphosyntactic Analysis and Parsing of Unrestricted Spanish Text. In: Proceedings of the First International Conference on Language Resources and Evaluation, Granada, Spain (1998)
4. Schmelkes, C.: Manual for the submission of blueprints and research reports. (Manual para la presentación de de anteproyectos e informes de investigación Tesis). Oxford, México (2006)
5. Dominguez S., A. P., & Gonzalez B., J.: Implementation of a parser for Spanish. Thesis (2002)

An Authoring Tool to Develop and Display Courseware in a 3D Learning Environment

Venustiano Soancatl[1], María-Luisa Cruz[1], Lucina Torres[1], Andrea Herrera[1], Luis David Huerta[1], Raúl Rodríguez[1], Antonio León[1], Inti Reyes[1], Nubia Cabrera[1], and Carmen Martínez[2]

{venus,mlcruz,luztorres,aismenee,luisdh2,raul.castillejos, leonborges,inti,csan}@bianni.unistmo.edu.mx
http://www.unistmo.edu.mx
cmartinez@unca.edu.mx
http://www.unca.edu.mx

Abstract. Authoring tools are becoming increasingly common, they make it easier and faster to create educational content. Normally, they are based on standard training practices, and the content usually is displayed in two-dimensions. A learning environment that contains a challenge manager to store exercises in a database has been developed as an authoring tool for a 3D educational game. The challenge manager allows teachers with no programming skills to design exercises and problems for the learning environment.

Keywords: Authoring tool, learning environment, courseware.

1 Introduction

An authoring tool is an application development environment for non-programmers, which has pre-programmed elements for the development of interactive multimedia software titles. According to Locatis [3], the term authoring tool refers to a range of software products having utilities for composing, editing, assembling and managing multimedia objects, whereas the term authoring system refers to a subset of these products allowing multimedia development without having to program.

These tools are classified from simple to advanced. The tool is simple when supporting utilities, for example, drag-and-drop facilities and wizard. Advanced tools require programming capabilities to build course material and need technical competency [2]. Some of them are unspecialized authoring tools, such as PowerPoint and Flash, FrontPage and Dreamwaver. For example, Wagner [8] uses PowerPoint as a scenario authoring tool in athletic training. Whereas examples of specialized authoring tools are Presenter, Engage, Quiz Maker, Course Lab and GLO Maker. Prensky [5] proposes to produce new types of training modules that are likely to draw interest from trainees in ways that the current authoring tools don't. For example, by trainees allowing orient themselves by walking around an accurate 3D representation to find clues and solve problems.

Courseware that is instructional material in an interactive mode facilitates and controls the individualized learning environment for students. It can be used to provide instructional material to a group of students or for the individual student. Courseware can be further subdivided into instructional methods, such as drill and practice, tutorials and problem solving which supplement or enrich the learning environment. The success of developing the courseware is governed by three major factors: the content and pedagogical quality of the learning materials, the amount and character of faculty support in the overall learning situation [1], and the motivational quality of the learning materials

Drill and practice is an author-controlled approach to develop courseware. The aim of this approach is to assume the main responsibility for developing the students skill in the use of a given concept. This involves leading the students through a series of examples where they can practice the material already learned or have it repeated. The assumption with a drill and practice system is that the students have already had the concept presented to them, that the material has been seen before, and the purpose now is to gain and develop familiarity with the ideas. Since the purpose of drill-and-practice is to increase learning effectiveness through repeated practice based on a stimulus-response theory of behaviorism, the frequency of repetition should have a direct effect on achievement. Research suggests that drill and practice-based lessons can be an effective means of teaching students of varying learning styles.

As with many approaches, courseware has some advantages and disadvantages. An advantage is the ability to individualize the instructional process so that multimedia content involves students actively in the learning process. It is impossible for the students to be passive in the situation, therefore the activity and their involvement facilitate learning. Courseware offers fast feedback so students are kept informed of their progress through immediate feedback, presentation and achievement summaries. The learning reinforcement is immediate and systematized. Instructions while developing courseware can be systematically prepared, sequenced, tested and revised. There is the possibility to create generic teaching strategies that can be used with different instructional content to represent abstract pedagogical entities and to design at the pedagogical level. For example: "give a hint" or "teach the prerequisites" [4]. An additional advantage is that courseware frees teachers for other necessary work and thereby increases educational productivity.

The major limitations to the widespread use of multimedia courseware and authoring systems in the educational systems are: 1. a lack of knowledge among the educators as how to effectively use the computer in an educational setting, 2. an insufficient quantity of high-quality courseware, which is closely related to the use of an inefficient authoring tool, 3. problems associated with the amount of time needed to develop materials and the difficulty of finding qualified and experienced instructional designers and computer programmers.

Some of the main features that an authoring tool should have are: 1) a user-friendly interface that projects usability, simplicity and ease of use. The user simply has to select the button to load the media object visible on the template

and use separate media editors to import the media files. 2) a database that is able to store, retrieve, update, sort and delete records. It makes it easy to improve content that has already been prepared for a specific use.

This paper describes authoring tools that can be used to easily develop courseware by teachers and domain experts with no programming skills by following the drill-and-practice approach. However, it also can be used for problem solving as described in [7]. Additionally, the content is presented in an attractive 3D learning environment to engage young students.

2 System Design

The 3 main components of the system are, a database, the Challenge Manager and the 3D learning environment. The database is shared between the Challenge Manager and the 3D learning environment, as shown in Figure (1). The Challenge Manager is mainly designed for teachers and domain experts, and the 3D environment for junior high or middle school students. The database stores information about the challenges, which can be questionaries, quizzes or step-by-step solved problems.

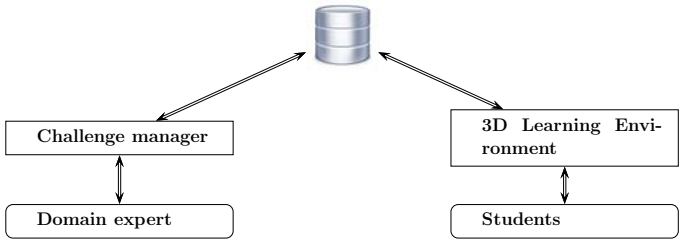

Fig. 1. System main components

2.1 Database

There are three main entities in the database: challenges, objects and users. A challenge is composed of a main statement, a figure linked to the statement, a time limit, a hint or recommendation, points, difficulty level, and a set of steps that lead the students to find the way to solve the challenge. A step is composed of a multiple choice question and if necessary a recommendation. A tip can be represented by text, a formula or an image. In this way, challenges can be questions or problems from any domain, such as math, history and even foreign languages. Challenges can be designed by teachers or domain experts and each challenge is linked to an object in the 3D environment.

Objects are fixed elements in the 3D environment and can be 2D images or 3D models. There are about 60 objects in the scenery which can not be

modified nor changed. Each object has a stored description and a position in the 3D environment. 2D images are famous philosophers or mathematicians and famous archeological places, such as, Teotihuacan, El Palenque and Machu Pichu. Additionally, architectural structures like the Big Ben in London and the Eiffel Tower in Paris are included to stimulate interest and interaction between the students and virtual environment.

In order to navigate in the virtual environment each user must have an account with a nickname and password. For each user, the database stores their name, age and gender. Also, it stores statistical information about the users such as the number of sessions in the system, how long the users use the system, number of conquered challenges, number of mistakes, number of attempts to conquer a challenge and scored points.

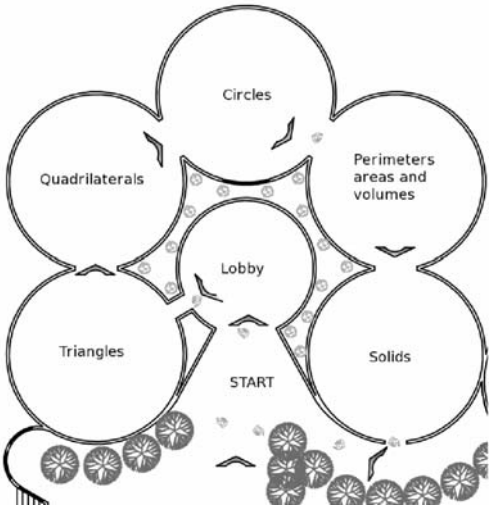

Fig. 2. Five main museum rooms

2.2 3D Learning Environment

The 3D learning environment is specially designed for students who like to play video-games focusing on the Game Based Learning approach as described in [6]. It is designed as a museum and the domain or main theme chosen to give examples is geometry. The museum has five rooms (Figure 2) and is called Geometry Virtual Museum (GVM). The GVM loads the challenges stored in the database and displays them as mentioned above. Each challenge is linked to an object and can be associated with multiple challenges. These objects can be images on the walls or 3D models such as platonic solids.

Figure (3) is a screenshot of the GVM showing elements taken from the database, such as pictures, the player nickname and score, the challenge, a

chronometer, the multiple choice question and the possible answers floating as balloons in the scenario. Users can navigate all around the GVM just as in a first person shooter game, but instead of a gun, users have a laser pointer and instead of projectile weapon-based combat, users interact with pictures and geometric challenges in the scenario. Every time users point on a picture, a short description of the picture is displayed on top of the screen. A click on a picture makes the system display a challenge step by step.

Fig. 3. Geometry Virtual Museum

The process of displaying a challenge, quiz or a problem is very simple, once a picture is clicked. The system displays the first challenge which starts with the main statement at the top of the screen and a chronometer begins. After a few seconds, the first multiple choice question is displayed at the bottom of the screen and choices are displayed in the 3D environment. If the correct choice is made by the user, the system displays a congratulating message and displays the next question (step). However, if the choice is incorrect, then a "try again" message is displayed on the screen. A recommendation can also be added to aid finding the correct solution. Also, to make the user think twice before making a wrong choice, they can be sent back to questions to refresh their memory. This process continues until the last multiple choice question is done, which increases the users' point score. Then the user can look for the next challenge to conquer. Additionally, when a challenge is displayed, it is stored in the database whether

it is overcame or not. Every mistake and the time taken to conquer the challenge is recorded for teacher reference. Thus, the question difficulty and effectiveness can be analyzed and adjusted.

2.3 Challenge Manager

The Challenge Manager (CM) allows teachers and domain experts to design instructional content to be included and displayed in the GVM. The process to design and integrate this content is very simple. The first step is to select the domain and five main themes, since the GVM has five rooms. For each main theme, it is possible to add as many subthemes and subsubthemes as the domain experts need. Figure (4) shows a schematic with themes and subthemes.

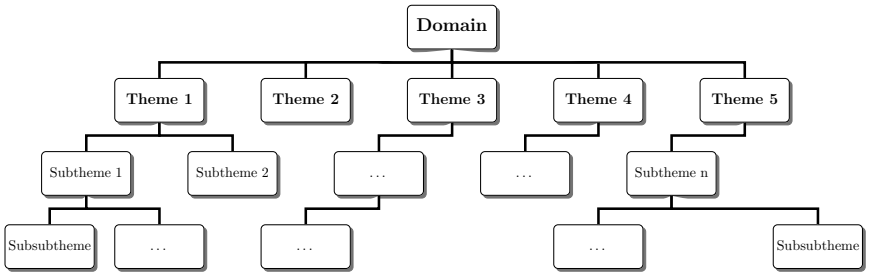

Fig. 4. Themes and Subthemes

The second step is to save the main theme names in the database using the CM. As mentioned above, each main theme corresponds to one room in the GVM. Every room has at least 10 objects on the walls. The third step is to design the challenges for each theme and subtheme, which can be quizzes or problems including multiple choice questions. The choices can be text or images. Challenges can be done using a simple text editor or a word processor. Figure (5) shows the structure of a challenge.

The fourth step is to add the challenges in the GVM using the CM. Since it has been designed to be used for non programmers, it is very easy to add instructional content into the GVM. It is necessary to choose one of the main themes saved in the second step, then the CM selects only the objects in that room. The domain experts can move through the objects and select one to link a challenge. Once an object is selected, it is possible to add a challenge to be displayed in the GVM.

Figure (6) shows a screenshot of the CM that is divided into three main vertical sections. The left section allows the domain expert to add five main themes and their subthemes. This allows them to select an object in the GVM in order to attach a challenge. The middle section permits the addition of a main statement of the challenge, a 2D image or figure associated to the challenge, a recommendation and points. It also permits the ability to update and delete

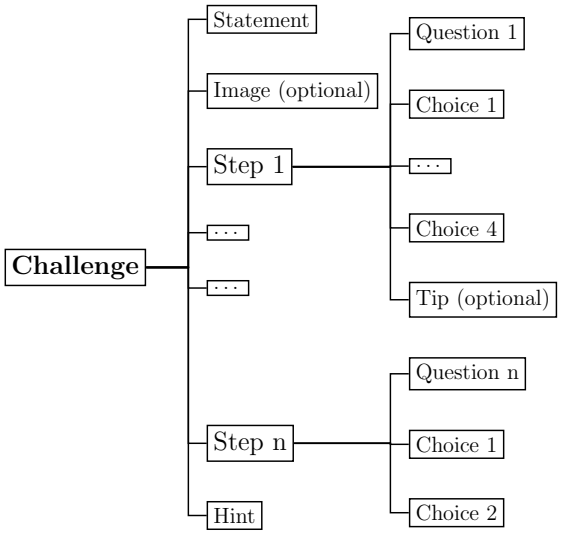

Fig. 5. General structure of a challenge

a challenge or to move through the challenges linked to the object. In order to delete a challenge from the database and GVM, it must not have any steps associated. The right section allows the addition of challenge steps (multiple choice questions) and a recommendation can be added if necessary. Choices can be text or images and for each challenge any number of steps can be added. It is recommended not to add too many steps since it can be tedious for the students. It is also possible to update and delete steps. The CM was developed using the programming language C# and SQL Server Compact Edition was used as the database manager.

2.4 Challenge Requirements

The design of challenges for the GVM has some requirements and limitations, for example, with the multiple choice questions, the first choice must be the correct answer. It makes it easier for teachers and domain experts to validate and review the correct answers and since choices in the GVM appear in a random order, students will not know the correct answer beforehand. The maximum number of words for a challenge statement is 40 or the maximum number of characters is 200. The maximum number of words for a step statement is 30 or the maximum number of characters is 125. The maximum number of characters for a text choice is 15. These text length restrictions allow the GVM to display text clearly in the scenery otherwise the text will be unreadable. The proportion between width and height or height and width of an image in the GVM should be 1.33 for aesthetic purposes.

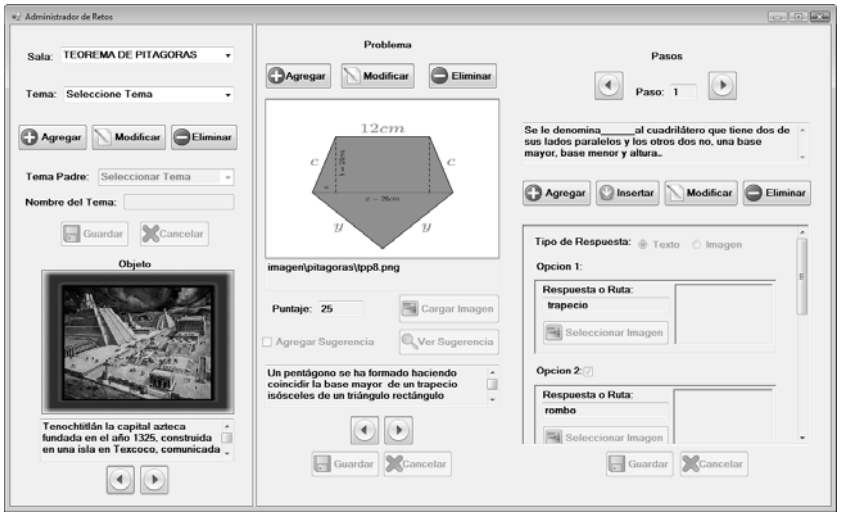

Fig. 6. Challenge Manager.

3 Examples

A possible challenge to be included in the GVM is the following: compute the volume of the building (Figure 7). Keep in mind that the side of each square is 5 meters.

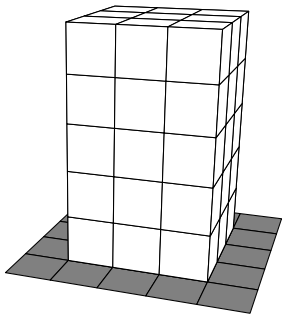

Fig. 7. Building

1. The formula to compute the volume of a quadrilateral prism is

(a) $V = L \times W \times H$ (c) $V = L^3$
(b) $V = L \times W$ (d) $V = W \times H$

TIP: Remember, the volume of a quadrilateral prism is the area of the base $\times$ the height.

2. The area of the base is

   (a) $225m^2$ (b) $125m^2$ (c) $325m^2$ (d) $225m^3$

3. The height of the building is

   (a) $25m$ (b) $15m$ (c) $25m^2$ (d) $45m$

4. The area of the base multiplied by the height is denoted by

   (a) 225×25 (b) 225×15 (c) 225×35 (d) 215×25

5. Solving the problem, the volume of the building is

   (a) $5625m^3$ (b) $5625m^2$ (c) $5625m^4$ (d) $5625m$

The above example is very simple and guides the student step by step through the process of solving the challenge, but more complex challenges can be designed. Additionally, different approaches can be used, for example, the problem solving Polya's method, question-led learning, action learning, learning by mistakes and others. The next example shows a challenge designed for non English speakers that can be included in the MVG

Challenge: Respond to the following questions about Chichén Itzá, which is one of the principal archeological sites on the Yucatán peninsula.

1. This is a photo of a

   (a) pyramid (b) house (c) monument

2. The base is a

   (a) square (b) circle (c) triangle (d) rectangle

3. How many sides does the pyramid have?

(a) four (b) three (c) forty

4. What is in the sky?

(a) clouds (b) rain (c) snow

4 Conclusions and Future Work

This paper describes an authoring tool that is designed to be non-technical and uses a familiar interface that increases usability. The tool is designed to provide assistance to teachers and students in putting together simple courseware. It displays content in a dynamic 3D learning environment where students can learn the content and get feedback immediately. Furthermore, domain experts can use multiple teaching strategies to develop courseware and the drill-and-practice is an approach that can be easily integrated in a courseware using this tool.

In the future we intend to add artificial intelligence to the GVM in order to increase flexibility and the speed in adjusting the exercises according to the users' responses and skills. By adjusting the level of difficulty and preventing the use of questions that are either too difficult or too easy, user interest will increase by avoiding boredom. Further work to the system will allow the reproduction of video and audio recommendation and will make the 3D learning environment work in network in order to allow many users surf at the same time.

Acknowledgements. We want to give many thanks to FOMIX-VERACRUZ and UNISTMO for supporting this work which is part of the project 95656. We also thank Kevin Mitchell for his contribution to the English review.

References

1. Ayub, M.N., Venugopal, S.T., Nor, N.F.M.: Development of multimedia authoring tool for educational material disseminations. Informatics in Education, Vol. 4, No. 1, pp. 5–18 (2005)
2. Khademi, M., Haghshenas, M., Kabir, H.: A review on authoring tools. In: Proceedings of the 5th International Conference on Distance Learning and Education, vol. 12. IACSIT Press, Singapore (2011)
3. Locatis, C., AI-Nuaim, H.: Interactive technology and authoring tools: A historical review and analysis. Educational Technology Research and Development, vol. 47 No. 3, pp. 63-75 (1999)
4. Murray, T.: Authoring knowledge based tutors: Tools for content, instructional strategy, student model, and interface design. Journal of the Learning Sciences. vol 7, No. 1, pp. 5–64 (1998)
5. M. Prensky: Modding - The Newest Authoring Tool. SRIC–BI report (2003)
6. Soancatl, V., Cruz, M.L., Huerta, L.D., Leon, A., Herrera, A., Torres, L., Zurita, W., Reyes, I.: Developing a virtual environment for learning geometry. Research in Computing Science, vol. 52, pp. 26-6 (2011)

7. Soancatl, V., Leon, A., Martinez, C., Torres, L.: Leading students to solve math problems using question-led learning. In: Meyer, B. (ed.) Proceedings of the 4th European Conference on Games Based Learning. pp. 368-374. Academic Publishing Limited. Copenhage (2010)
8. Wagner, R.: Using computer-based scenario authoring tools in athletic training. Athletic Training Education Journal, vol. No. 1, pp. 40-44 (2010)

Architecture for an Intelligent Tutoring System that Considers Learning Styles

María Lucila Morales-Rodríguez, José Apolinar Ramírez-Saldivar,
Arturo Hernández-Ramírez, Julia Patricia Sánchez-Solís,
and José Antonio Martínez-Flores

Instituto Tecnológico de Ciudad Madero, División de Estudios de Posgrado e Investigación,
Ciudad Madero, Tamaulipas, México

{lmoralesrdz, jpatricia.sanchez, jose.mtz}@gmail.com,
apolinar_r@yahoo.com, ahr@prodigy.net.mx

Abstract. In this paper we propose the architecture of an Intelligent Tutoring System that considers the student's learning style and the competency-based education. We also describe the processes that have been implemented so far. Our architecture presents innovations in the representation of the tutor module and in the knowledge module; the tutor module incorporates a selector agent, which will choose the content to show, considering the teaching strategies that support the student's learning style.

Keywords: Intelligent Tutoring System, learning style, teaching strategies.

1 Introduction

The idea of applying IT tools to teaching goes back to the 50's, but it wasn't until the 80's when computer-assisted teaching regained a special interest due to the techniques of Artificial Intelligence. At that time, the systems called Intelligent Tutoring Systems (ITS) arose, with the aim of developing the processes of education adapted to the different users/students [1].

An Intelligent Tutoring System (ITS) provides learning and / or customized training to students [2]. The personalized training is an argument on behalf of these systems and it is supported by the analysis realized by Bloom [3]. This analysis discusses the importance of adapting education to each student, confirming the individualized instruction as the most effective way of learning.

In general, an ITS is composed by three modules: the student module, the knowledge module and the tutor module [4]. Nowadays ITS are called Learning Management Systems (LMS) [5]. In our project, the LMS Moodle will be used.

The purposes of this paper are: a) to present the integration of an intelligent agent in the tutor module of a LMS, b) to describe the implementation of modules that have been developed.

This paper is structured as follows: section 2 describes the concepts related to intelligent tutoring systems. Section 3 deals with issues of learning styles and teaching strategies. Section 4 presents the proposed architecture. Section 5 describes the processes implemented and section 6 presents conclusions and future work.

2 Intelligent Tutoring Systems

Guardia (1993) cited by Salgueiro [6], presents a definition for the intelligent tutors: "*An ITS is a system of computer-assisted instruction, which uses Artificial Intelligence techniques, mainly for representing knowledge and drives a teaching strategy; and it is able of behaving like an expert, both in the knowledge domain that it teaches (showing the student how to apply that knowledge), as in the pedagogic domain, where it is able of diagnosing the situation in which the student is and offer a solution that allows him/her progress in the learning"*.

Within the literature reviewed, we found that areas like Mathematics, Physics and Programming, had implemented the basic architecture of an Intelligent Tutoring System. Among the related works we can cite Butz [7], Conati [8], Graesser [9] and Melis [10].

According to Butz [7], the basic architecture of an ITS is composed by a *student module*, a *knowledge module* and a *tutor module* which is also called teaching strategies module. These modules operate interactively and communicate through a central module, which it is often called *user interface*. This architecture is shown in Figure 1. Its modules are described below.

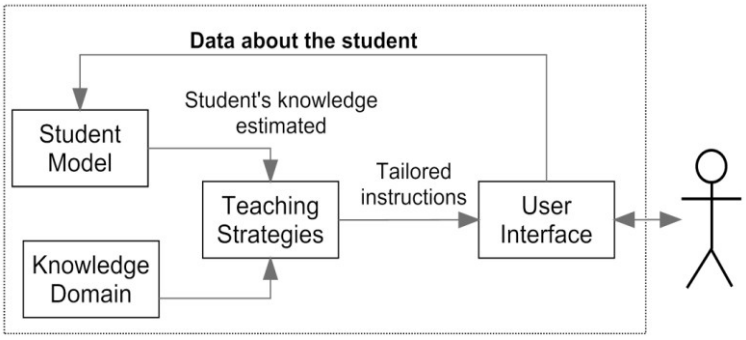

Fig. 1. Basic architecture of an ITS [7].

The student module aims to perform the student's cognitive diagnosis and the student's representation for future system feedback. Cataldi [11] proposes to incorporate learning styles in the ITS. According to her, the student module is composed by the following components:

- A database with learning styles available in the system.
- A map of the knowledge obtained initially from the domain module, which will be modified by the update of knowledge, based on the assessments made by the tutor module.

The knowledge module aims to store the dependent and independent knowledge of the scope. Basically, this module is composed by [11]:

- *Knowledge:* It refers to the content that must be loaded into the system, through the concepts, questions, exercises, problems and their relationships.
- *Didactic elements:* they are multimedia material, i.e., images, videos and sounds that help the student obtain knowledge during the teaching session.

The tutor module defines and implements a pedagogical teaching strategy, contains the objectives to be achieved and the plans to achieve them. This module selects the exercises, monitors the performance, provides assistance and selects the learning material for the student. It consists of the following sub-modules [11]:

- *Lesson Planner that* organizes the lessons' contents.
- *Profile analyzer,* which analyzes the characteristics of students, selecting the most appropriate pedagogical-teaching strategy.

The user interface specifies and provides support to the students' activities and to the methods used to perform these activities. The interface should be easy to use and attractive. Thus, the students quickly learn how to use it, and they can focus all their attention on the process of learning the subject [12].

3 Learning Styles and Teaching-Learning Strategies

In [13], the fact that each person uses their own methods or strategies to learn is called "learning style". Teaching strategies are procedures or resources used by the teaching agent to promote meaningful learning [14]. Although the strategies vary depending on what you want to learn, each person tends to develop certain preferences or global trends that define a style of learning [13].

According to [13], the notion that each person learns differently from others, must be considered to facilitate their learning, however, we must be careful not to "label" people, since the learning styles, although relatively stable, change depending on the situation and are susceptible to improvement. Moreover, Gomez [13] states that when students are taught according to their own learning style, they learn more effectively. Some of the most known and used learning styles models are:

- Felder and Silverman model.
- Kolb model.
- Neurolinguistic Programming model of Bandler and Grinder.
- Multiple Intelligences model of Gardner.
- VARK model of Neil Fleming.

Some learning styles have been considered in the implementation of tutoring systems in order to adapt their environments of teaching toward users. For example, the Richard Felder learning style model is considered in the systems presented by Hernandez [15] and Caviedes [16]. In the system proposed by Cataldi [17], in addition to the Felder model, they also considered the Multiple Intelligences model of Howard Gardner. Finally Araújo [18] and Peter [19] used the VARK model of Neil Fleming.

4 Architecture Proposed for an ITS

This paper proposes to incorporate to the classic architecture of an Intelligent Tutoring System, a process that selects the contents to show, influenced by the teaching strategies that encourage the student's learning style. These teaching strategies will be the link to select the contents of the subject. The contents should be prepared for each one of the teaching strategies for each learning style.

Figure 2 shows the proposed architecture, in which we specified the added components with a highlighted line. We also present a paused line that represents the redesigned component of the general architecture of an ITS and a dotted line that represents the components used for the Learning Management System.

Below we describe the changes made to the modules of the general architecture of the ITS.

In the *tutor module* we incorporated the teaching-learning strategies considered in the design of the themes of the subject, as well as the redefinition of teaching strategies according to student's learning style. We also incorporate 2 processes to adapt the contents to be presented: 1) that selects the topics to show to the students by linking their learning style with teaching strategies used in the creation of the topics, thus support their learning, and 2) a process for the diagnosis of competencies.

In the *knowledge module* we added a corpus that will store the competencies of the subject and some metadata to label the contents of the subject and characterize the competencies to develop.

In the interface module we incorporate a filter to show the contents chosen by the selection process of the tutor module.

The proposed architecture process is shown in Figure 3. The process begins determining the student's learning style. To achieve this, we decided to use the VARK model of Neil Fleming [20], which uses a questionnaire to identify the student predominant style. This questionnaire will be applied once, when the student initiates the session for the first time in the platform.

Once the student's learning style is determined and stored, the student's diagnose process (competencies) begins. The student's diagnose process doesn't take any action when students enter the system for the first time, because there is still no information to be processed.

The next step is to choose the items to be displayed by the selector agent, which must be designed in conformity with a teaching-learning strategy according to the student's learning style. The agent is also responsible for readjusting (increasing) the exposure time of the subjects based on the following: a) if the subject is elaborated in

the English language and the student doesn't fulfill the level required to understand the English language, and b) if the item to be displayed isn't elaborated under any teaching strategy to support the student's learning.

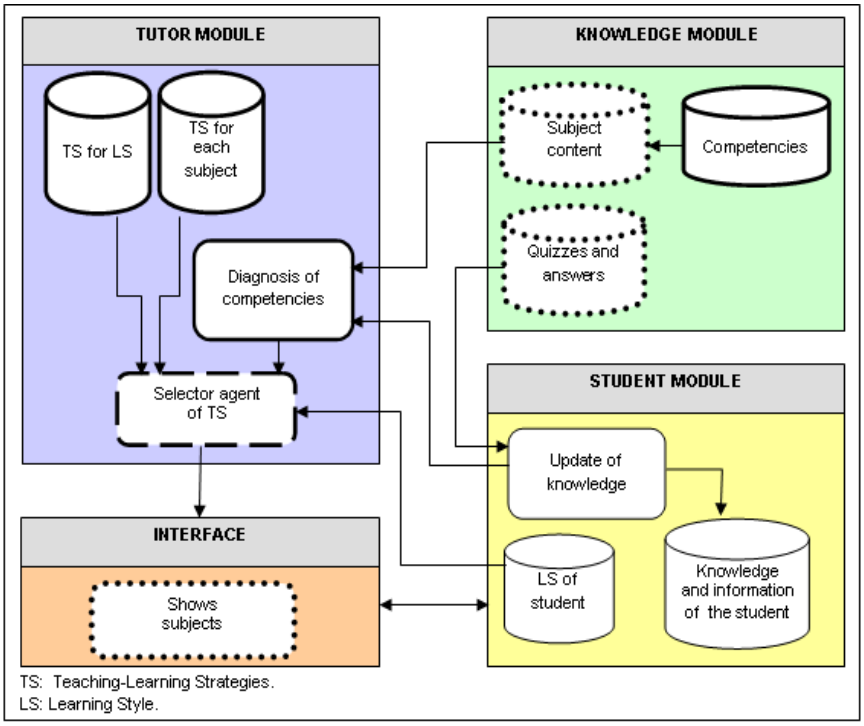

Fig. 2. Proposed architecture.

Since the subjects have been selected and personalized, these will be shown to the student through the Moodle platform. Once the student has revised a subject, its evaluation will be applied and the database that stores the student's performance will be updated with the results obtained from the evaluation.

On the other hand, when the student's diagnose process has historical information of him/her, the process will take the information of the student's performance (elapsed time of the activity, amount of attempts and assessments) and it will evaluate if the student gets the competency (Bayesian network), if so, the student will continue to the next topic and the selector agent will not execute any action. Otherwise, the selector agent will select another resource to display, elaborated with a different teaching strategy that supports the student's learning style.

5 Implemented Processes

The implementation of the VARK questionnaire is part of the process of the proposed architecture, see Figure 3. This implementation will help to determine the student's learning style. The following sections describe its development.

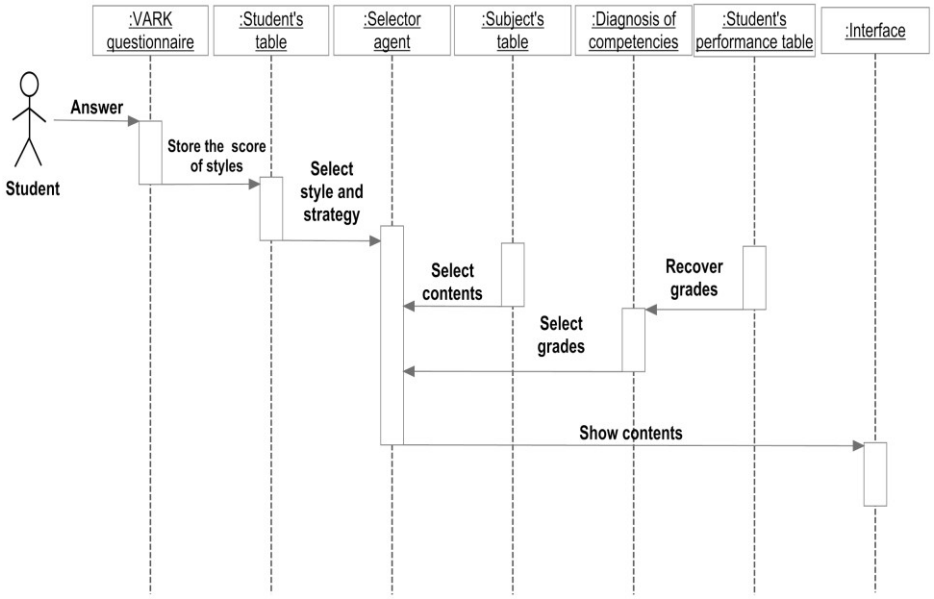

Fig 3. Process sequences of the proposed architecture for an intelligent tutor system.

5.1 VARK Model

The VARK model was developed by Neil Fleming [20] in 1987. This model is a tool to determine the preferences of individuals for information processing. The model considers four different learning styles: Visual, Aural, Read / Write and Kinesthetic.

The model consists of a questionnaire with 16 questions and each one has four answer options that correspond to each one of the learning styles. To answer the questionnaire, it is necessary that the student: choose the answer which best explains his/her preference and mark the letter(s) that represents his/her choice. The student can select more than one response to a question if one does not fit with his/her perception. The student can also leave a blank if any question isn't applied to his/her preferences, but the student must answer at least 12 questions.

Table 1 shows some of the Neil Fleming's proposals for teaching-learning strategies for each learning style.

5.2 Implementation of the VARK Questionnaire in Moodle

The integration of the VARK questionnaire in Moodle allows us to determine the student's preferred learning style. The following section describes the four activities for its implementation.

Table 1. Teaching-learning strategies for the VARK model.

Learning Style	*Teaching-Learning Strategies*
Visual	▪ Pictures, Videos, Posters.
Aural	▪ Discuss topics with your teachers. ▪ Explain new ideas to other people. ▪ Use a tape recorder.
Read / Write	▪ Dictionaries, Textbooks, Notes.
Kinesthetic	▪ Field tours. ▪ Applications. ▪ Trial and error.

5.2.1 Configuration in Moodle of the student's custom fields to store the questionnaire results

Moodle allows customizing fields to the student's profile, which was used for adding the fields: Visual, Aural, ReadWrite and Kinesthetic. These fields were used to store the results obtained from applying the VARK questionnaire to the student. To add these fields the following steps were done:

Step 1. Log in to the Moodle platform with an administrator's account, access the block Administration of the site and choose the options: *Users* > *Accounts* > *User profile fields*.
Step 2. Create a new custom field of text type.
Step 3. Configure the field as blocked and left empty the default value option.

This same procedure was used to add the *Actual Style* field, which will store the learning style resulting from the application of the questionnaire to the student. Also, the ActualStrategy field was added for storing the actual teaching strategy.

5.2.2. Presentation of the questionnaire to the student

Due to the fact that the VARK questionnaire must be applied and stored only once, it was necessary to detect in Moodle's code, the point where the student's login is validated (index.php file, located in the moodle\login folder). In this file we added a consult of the custom fields values: Visual, Aural, ReadWrite and Kinesthetic. If these values are empty this indicates that VARK questionnaire was not applied, so the web page that contains the questionnaire will be shown. The code added to the index.php file is:

```
if ($USER->Visual == "" and $USER->Aural == "" and
    $USER->ReadWrite == "" and
    $USER-> Kinesthetic == "")
     {redirect($CFG->httpswwwroot.'/login/p_vark_preguntas.php');}
```

Custom fields Visual, Aural, Read-Write and Kinesthetic, ActualStrategy and Actual_Style are automatically loaded for the entire student session in the global object $USER, which gives us access to them.

When the values of the student custom fields are empty, the next line will be executed: redirect ($CFG-> httpswwwroot. '/ Login / p_vark_preguntas.php'), which redirects to the VARK questionnaire, otherwise the VARK questionnaire will not be displayed.

5.2.3. Implementation of the questions and answers of the questionnaire

To integrate the VARK questionnaire the following files were incorporated into Moodle: *p_vark_preguntas.php*, *p_vark.html* and *p_vark.php*, which were located in the folder moodle \ login.

The *p_vark_preguntas.php* file incorporates as part of its code, the contents of the *p_vark.html file*, which has the code to display the questions and answers of the questionnaire. Figure 4 shows the VARK questionnaire web page. The *p_vark.php* file is responsible for the evaluation of the questionnaire and it stores the results of the evaluation in the custom fields.

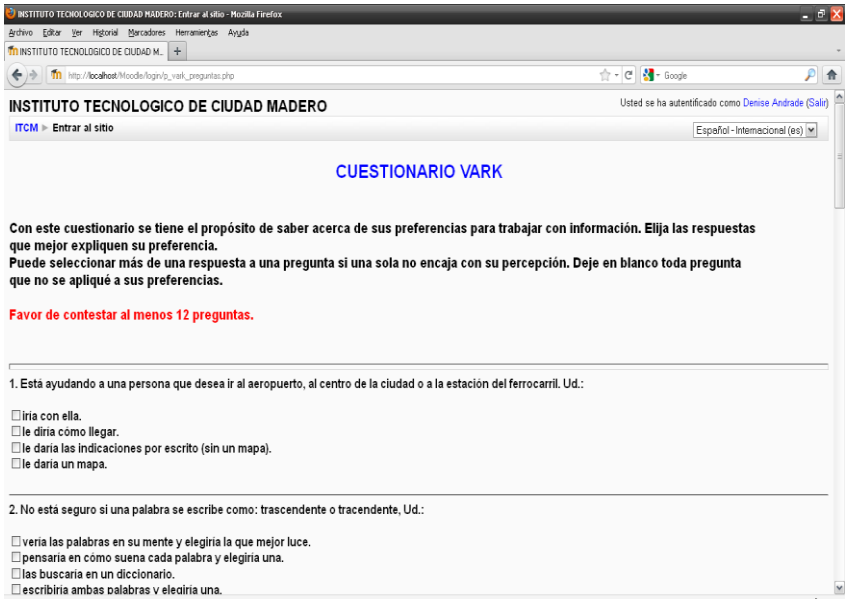

Fig. 4. Web page showing the VARK questionnaire.

5.2.4. Evaluation and storage of the results of the questionnaire

The process of obtaining the results of the VARK questionnaire considers three steps, which are detailed below:

1. To have a table of predefined answers for the model [20]. A fragment of this is shown in Table 2.

Table 2. Fragment of the table of predefined answers.

Question	*Letter a*	*Letter b*	*Letter c*	*Letter d*
1	K	A	R	V
2	V	A	R	K
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
15	K	A	R	V
16	V	A	R	K

2. Identify and mark in the table the letter(s) that corresponds to the answer(s) selected, for example, if the student chose the letters **b** and **c** in question 1, he/she must mark the letters **A** and **R** in the row in question 1, see Table 3.

Table 3. Identification of answers.

Question	*Letter a*	*Letter b*	*Letter c*	*Letter d*
1	K	A	R	V

3. Summarize each of the letters V, A, R, K chosen in each question. The letter that has obtained the major number of points will be the one that indicates the learning style preferred by the student. In case of tie between two or more letters, a *Multimodal* learning style is considered.

The previous steps were implemented in the file *p_vark.php*. This file calculates the punctuation for each letter V, A, R, K and stores the information on the custom fields: Visual, Aural, ReadWrite and Kinesthetic. Also, it stores in the custom field, Actual_Style, the learning style with the highest score. And finally, it is responsible for selecting and storing in the custom field ActualStrategy one of the teaching strategies associated with the current learning style.

6 Conclusions and Future Work

In this paper we propose an extension of the classical architecture of an Intelligent Tutoring System, which incorporates a selector agent of contents. This selector agent, in its selection process, considers the teaching strategies that support the student's learning style. This extension will be implemented using the Moodle LMS.

Also we described the process of how to integrate the VARK questionnaire in Moodle to obtain the student's learning style.

The proposed architecture is currently in development. The VARK model implementation was the first step of the project, leaving the selector agent implementation to display the subject content and the competency evaluation process as future work.

References

1. Urretavizcaya, L. M.: Sistemas Inteligentes en el ámbito de la Educación. Revista Iberoamericana de Inteligencia Artificial, volumen 5, Num. 12, pp. 5-12 (2001)
2. VanLehn, K.: The Behavior of Tutoring Systems. International Journal of Artificial Intelligence in Education, 16:227-265 (2006)
3. Bloom, B. S.: The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-one Tutoring. Educational Researcher, Vol. 13, No. 6, pp. 4-16 (1984)
4. Cataldi, Z., Lage, F. J.: Sistemas Tutores Inteligentes orientados a la enseñanza para la comprensión. EDUTEC Revista electrónica de Tecnología Educativa, No. 28 http://edutec.rediris.es/Revelec2/Revelec37/ (2009)
5. Cárdenas, P. J. L.: Sistemas de Gestión del Aprendizaje, SGA (LMS). Facultad de Educación, Universidad Autónoma de Yucatán, México.
6. Salgueiro, F. A.: Sistemas Inteligentes para el Modelado del Tutor. Tesis de grado en Ingeniería Informática, Facultad de Ingeniería, Universidad de Buenos Aires, Argentina: 196p. (2005)
7. Butz, C. J., Hua, S., Maguire, R. B.: AWeb-based Bayesian Intelligent Tutoring System for Computer Programming. Department of Computer Science, University of Regina (2006)
8. Conati, C.: Intelligent Tutoring Systems: New Challenges and Directions. In Proc. of the 20th International Joint Conference on Artificial Intelligence, pp. 2-7 (2009)
9. Graesser, A. C., Person, N. K., Harter, D. and The Tutoring Research Group.: Teaching Tactics and Dialog in AutoTutor. International Journal of Artificial Intelligence in Education, vol. 12, pp. 257-279 (2001)
10. Melis, E., Siekmann, J.: Activemath: An intelligent tutoring system for mathematics. Seventh International Conference 'Artificial Intelligence and Soft Computing' (ICAISC), 3070:91–101 (2004)
11. Cataldi, Z., Lage, F. J.: Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia. Laboratorio de Informática Educativa y Medios Audiovisuales, Facultad de Ingeniería, Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, Argentina, (2009)
12. Millán, D. E.: Sistema bayesiano para modelado del alumno. Tesis doctoral, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España (2000)
13. Gómez, N. C. L., Aduna, L. A., García, P. E., Cisneros, V. A., Padilla, C. J.: Manual de Estilos de Aprendizaje. Secretaría de Educación Pública, Dirección General del Bachillerato,http://www.dgb.sep.gob.mx/informacion_academica/actividadesparaescolares/multimedia/Manual.pdf (2004)
14. Díaz, F. D., Hernández, G.: Estrategias docentes para un aprendizaje significativo. Capítulo 5, McGRAW-HILL, pp.13-28 (1999)
15. Hernández, Y., Rodríguez, G., Arroyo-Figueroa, G.: Integrating learning styles and affective behavior into an intelligent environment for learning. Research in Computing Science 51, pp. 172-180 (2010)

16. Caviedes, P. D., Medina G. V. H., García P. O.: Diseño de un sistema tutor inteligente basado en estilos cognitivos. Realizado en la Maestría de Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas en Bogotá – Colombia (2009)
17. Cataldi, Z., Lage, F. J.: Modelado del Estudiante en Sistemas Tutores Inteligentes. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, No. 5, Art. 4, (2010)
18. Araújo, G. A., Miranda, R., Almeida, Z., Rocha, F. L.: A Web-Based Intelligent Tutoring System on Teaching and Learning Electrical Project. Instituto Superior de Engenharia do Porto, Porto, Portugal (2010)
19. Peter, S. E., Bacon, E., Dastbaz, M.: Learning styles, Personalisation and Adaptable e-Learning. Fourteenth International Conference on Software (2009)
20. VARK: a guide to learning styles, http://www.vark-learn.com/english/index.asp (2011)

Survey on Understanding the Tutorial Actions based on Students' Affect

Yasmín Hernández[1], Gustavo Arroyo-Figueroa[1], and L. Enrique Sucar[2]

[1] Instituto de Investigaciones Eléctricas, Gerencia de Sistemas Informáticos
Reforma 113, Col. Palmira, 62490, Cuernavaca, Morelos, Mexico

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Coord. Ciencias Computacionales
Luis Enrique Erro # 1, Tonantzintla, Puebla, Mexico

{myhp, garroyo}@iie.org.mx, esucar@inaoep.mx

Abstract. We have developed an affective behavior model for intelligent tutoring systems that considers both the affective and knowledge state of the student to generate tutorial actions. The affective behavior model was designed based on teachers' expertise obtained through a survey which 11 math teachers participated. The study focused in knowing how teachers manage the affective state of the students in order the students learn. During the survey, teachers watched a video of a student interacting with an educational game with an animated pedagogical agent. We asked them which agent's animation and which pedagogical actions are suitable for affect and knowledge of the student in each student's movement.

Keywords: Pedagogical agents, student affect, teachers' expertise, intelligent tutoring systems.

1 Introduction

Emotions have been recognized as an important component in motivation and learning. There is evidence that experienced human tutors monitor and react to the emotional state of the students in order to motivate them and to improve their learning process [5]. Recently there has been extensive work on modeling student emotions in intelligent tutoring systems, see [1]; however, there have been only limited attempts to integrate information on student affect in the tutorial decisions, e.g. [9, 2, 8]. If we want to consider the student affective state in the tutorial actions, an important problem is to identify the best tutorial action given both the students' affective and knowledge state. We conducted a survey consisting in to interview qualified teachers aimed at understanding which actions the teachers select according to the state of a student's affect and knowledge. The results are being used to develop an affective behavior model that considers both the affective and knowledge state of the student to generate tutorial actions. In this paper we describe the survey and our findings.

2 The Prime Climb Educational Game

To conduct the survey we use Prime Climb, an educational game to learn number factorization; t his game includes a pedagogical agent with a model of student's knowledge [6]. In Prime Climb, two players have to climb mountains in a collaborative way. Each mountain is composed by hexagons labeled with numbers. Players have to move to a number that does not have common factors with the partner's number, if not they fall off the mountain. To give adequate instruction, Prime Climb relies on a Bayesian pedagogical student model. The student model assesses the evolution of a student's factorization knowledge during interaction with the game. The pedagogical student model is used by an animated agent to deliver hints when it has evidence that the student is not learning from the game. The animated pedagogical agent is implemented through the Merlin character of Microsoft agent [7].

3 The Affective Model

Once the affective student state has been obtained, the tutor has to respond accordingly. The tutor needs an affective model which establishes parameters that enable a mapping from the affective and pedagogical student state to tutorial actions. The tutorial actions are composed by a pedagogical action and an affective action. The affective action tries to promote a positive affective student state and the pedagogical action to convey knowledge the student needs to know.

We consider as affective actions the way in which the pedagogical content is delivered to the student; e.g., the words, the facial expression, colors or sound included in the message. In the work presented here, an affective action is an animation of a pedagogical agent who delivers the pedagogical actions to students. In this way, the tutorial action is composed by an affective action and a pedagogical action.

Our main hypothesis is that the tutor action has a direct influence on learning and on the affective state of the student; and by selecting the appropriate tutorial action (i.e. according to the current student state), the tutor could improve the learning process and the affective state of the students. Given this hypothesis, we want to help students to learn and at the same time to foster a positive affective state.

4 The Teachers Survey

We conducted a survey with skilled teachers to validate our assumptions and refine our model. We wanted to know which actions the teachers do according with the affective and pedagogical student state and why they select those actions. Eleven math teachers participated in the survey, they have taught by 17.63 years in average from high school to post grade. These teachers have been trained in several teaching methodologies.

The survey consisted in to have teachers watching a video of a student interacting with Prime Climb and to ask them to say which affective and pedagogical actions and why they shall do in order to help student to learn.

The survey consisted in 1) we explained the teachers the aim of this study, and our main motivations, and hypothesis; 2) the teachers interacted freely with Prime Climb to familiarize themselves with the game; 3) teachers were shown the Microsoft agent animations, and were asked to say which animations they considered suitable to provide affective tutorial feedback as affective action in Prime Climb; 4) The teachers viewed a video of the interaction of one student with Prime Climb and were asked to say which affective and pedagogical actions they shall do according to specific student states and tutorial situations; 5) Teachers answered three general questions about the relationship between affect and teaching. Each complete teacher's session lasted 90 minutes approximately.

Firstly, we explain teachers the context of the survey, we explained what "affective action" is our work, and we want to use the Merlin's animations as affective action trying to promote a positive affective state. We explain that the affective action in conjunction with a pedagogical action compose a tutorial action to be delivered to students. Then, the teachers interacted with Prime Climb as much time as they wanted to familiarize with the environment and to see different situations could present in a student interaction.

After that, teachers were shown the Microsoft Agent animations with the character Merlin (Microsoft Agent Merlin Character is a copyrighted work of Microsoft Corporation), and were asked to say which animations they considered suitable to provide affective tutorial feedback in Prime Climb. The character Merlin supports 73 animations some examples are listed in the table 1.

Table 1. Examples of the character Merlin of Microsoft Agent. It is listed the name of the animation and adescription of what the agent do when the animation is played.

Merlin´s Animation	Animation Description
Decline	Raises hands and shakes head
DontRecognize	Holds hand to ear
Process	Stirs caldron
Read	Opens book, reads and looks up
Search	Looks into crystal ball
Suggest	Displays lightbulb
Sad	Sad expression
Think	Looks up with hand on chin
Wave	Waves

The aim of this phase is teachers could see the potential of the animated agent and they could select the animations they wanted to use in the survey next phase, but if they wanted they could have available the complete animations. Two professors wanted to have all the animations available they said they did not know the situations they will find and they could discard any animation.

The teachers selected the animations that they deemed to be generally appropriate to convey affective elements via a program; in Fig. 1 we show a screenshot of the

program used in this phase; they could select any animation they want the animated agent perform as many times as they wanted.

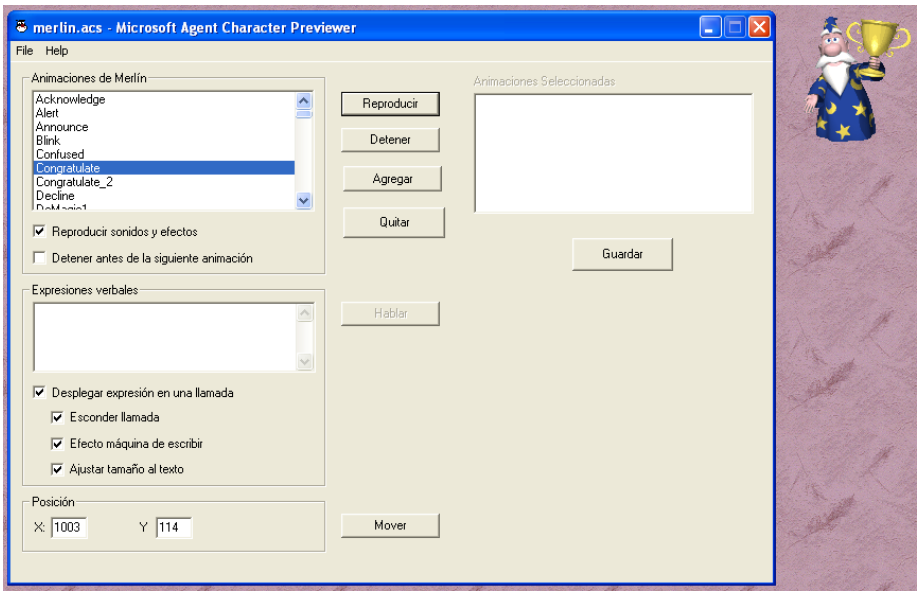

Fig. 1. Screenshot of the program used to see the character and animations of Microsoft Agent (Mostly in Spanish). In the combo list on the left they can select the animations to be play by the character, and when they found an interesting animation they can select it to add it to the combo list on the right. The agent is playing "congratulate" animation.

Subsequently, the teachers viewed a video of the interaction of one student with Prime Climb. The interaction lasted approximately five minutes, during this time the student climbed three mountains (levels). This specific video was selected because it showed a variety of tutorial situations based on a mix of student's correct and incorrect behaviors. While it would have been more principled to show the teachers interactions of several different students with Prime Climb, this was not possible because of constraints on the teachers' availability. Fig. 2 shows a screenshot of the program use in this phase of the study.

Teachers were provided with facilities to stop and replay the video as many times as they wanted. After each student's move, they were asked to rate the student's affective state and to establish the pedagogical and affective components of the tutorial action that they considered adequate at that particular point. We also asked teachers to say how they thought the selected action improved the student's affect and knowledge. An example teacher's report is presented in Fig. 3.

This phase of the study is very important because it provides information about how the teachers choose their actions considering the affective and the knowledge states of the students; we assume that teachers selected actions that they believed would improve a student's affective state and knowledge.

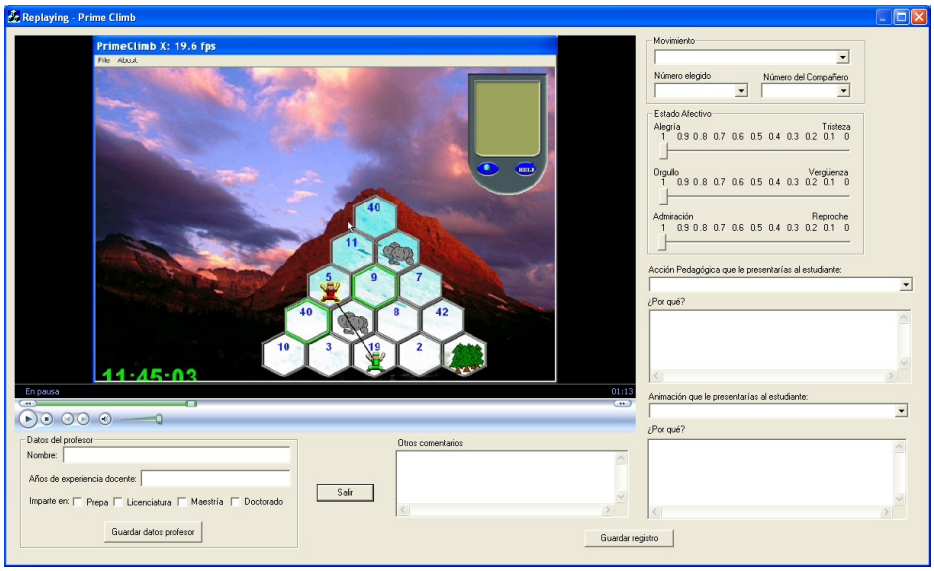

Fig. 2. Interface of the program for the teachers' survey (in Spanish). Left: Video of a student interacting with Prime Climb. Right and bottom: options for the teachers. The teachers were asked to establish the affective and pedagogical action to be presented to students taking into account the affective and knowledge state, according to student performance in the video.

Affective state:	**Pride/Shame**	75/25
	Admiration/Reproach	70/30
	Joy/Distress	73/27
Knowledge state:	Student knows the numbers factorization	
Pedagogical action	Right, these numbers do not share factors	
Affective action	Congratulate_2	
Pedagogical action explanation	Student made a correct click	
Affective action explanation	Student is having success	
Comments	I try to motivate the student	

Fig. 3. Example of a teacher's report from the second phase of the teachers survey. The teachers said what they should do and why according to the student state.

We also want to know more about the relationship between affect and teaching; therefore at the end we asked teachers the next three questions: 1) Do you take into account the students' current knowledge and affective state when you are teaching? Why? 2) Which is more important for you, knowledge or affect? Why? and 3) Can you group your actions into some categories?

5 Results

From the complete group of animations some of them were not used because those animations compose an animation loop, for example, "read" and "continue reading",

therefore we had 58 animations to evaluate. In this phase we obtain 53 animations was selected at least once, 46 animations was selected at least twice, and 17 animations was selected more than five time. In the Table 2, we present the animations selected more than 5 times.

Table 2. Animations selected by the teachers. It is listed the animations selected more than 5 times (only five animations was not selected by the teachers).

Animation	Times it was selected
Confused	9
Congratulate_2	9
GetAttention	8
Hide	8
Read	8
Decline	7
Suggest	7
Announce	6
Congratulate	6
MoveDown	6
MoveLeft	6
MoveRight	6
MoveUp	6
Pleased	6
Process	6
Search	6
Show	6

Considering the previous, we believed Merlin is a character with the suitable expressivity to present the tutorial actions and it can be used in an educational environment, since more that 90% of its animations deemed suitable to the teachers.

In the next phase, the teachers mapped student states to affective actions. Based on the teachers' responses, we selected 14 of the 58 animations as those most potentially effective as affective components of Merlin's interventions. These 14 actions are listed in Table 3.

Finally, we asked teachers the next three questions: 1) Do you take into account the students' current knowledge and affective state when you are teaching? Why? 2) Which is more important for you, knowledge or affect? Why? and 3) Can you group together your actions into some categories?. The answers to these questions are presented in Table 4.

In the third question, we asked teachers to try to categorize their actions into some categories. The answers to this question were general and open, therefore it was difficult to obtain a teachers' actions classification; however, all the participating

teachers stated the aim of their actions is to motivate students, and the last aim is student learning. Some the categories mentioned by the teachers are in Table 5.

Table 3. Merlin's animations selected as affective action in a tutorial action.

Affective action	Animation Description
A1-Acknowledge	Nods head
A2-Announce	Raises trumpet and plays
A3-Congratulate	Displays trophy
A4-Congratulate2	Applauds
A5-DoMagic1	Raises magic wand
A6-DoMagic2	Lowers wand, clouds appear
A7-Greet	Bows
A8-Hide	Disappears under cap
A9-Pleased	Smiles and holds hands
A10-Alert	Straightens and raises eyebrows
A11-Confused	Scratches head
A12-Explain	Extends arms to side
A13-GetAttention	Leans forward and knocks
A14-Surprised	Looks surprised

Table 4. Answers to questions: Do you take into account the students' current knowledge and affective state when you are teaching? Which is more important for you, knowledge or affect?

Description	Times/%
Teachers who take into account only the students' knowledge	1/11 (9%)
Teachers who take into account only the students' affect	1/11 (9%)
Teachers who take into account both the students' knowledge and affect	9/11 (82%)
Teachers who think the students' knowledge is more important	4/11 (36%)
Teachers who think the students' affect is more important	4/11 (36%)
Teachers who think both states are important in the same way	3/11 (27%)

Table 5. Answers to question: Can you group together your actions into some categories?

Categories
Positive feedback
Negative feedback
Reward
Reprimand
Motivation
Get attention
Relax situation
Harder exercises

We used the teachers' reports to establish the probabilities describing the impact of the various affective and pedagogical components of an action on knowledge and affect, given the current student's state and outcome of student's action. These are the probabilities used by a dynamic decision network in the affective model to calculate the expected utility of actions. For example, when a student made a successful move but seemed not to know the numbers factorization, teachers often selected the verbal hint "You're right again! But do you know why? Here's an example", where the example is an explanation about the factorization of the relevant numbers. Thus, the CPTs describing the factorization knowledge of the numbers involved in a student's correct move at next time are set so that, if the knowledge is predicted to be low at current time.

6 Conclusions and Future Work

We present a survey we conducted to know what teachers to according with affect and knowledge of student in order students learn, we presented the results. We believed the results are encouraging because they show what teachers do when they are teaching, we need more data to have stronger findings. We use the results to build a model and conducted a user study to evaluate the affective behavior model, showing that for younger students there is positive impact on learning [3, 4]. We want to conduct another study, having more students interacting with the model, and in this way to complete the integration of the affective model with an educational environment.

Acknowledgments. We would like to thank Cristina Conati for many useful discussions on the definition of the affective model. This research was supported by the *Instituto de Investigaciones Eléctricas*, Mexico.

References

1. Conati, C., Mclaren, H.: Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction, 19, 267-303 (2009)
2. Faivre, J., Nkambou, R., Frasson, C.: Toward Empathetic Agents in Tutoring Systems. In: Russell, I., Haller, S., FLAIRS'03, pp. 161-165. AAAI Press, Florida (2003)
3. Hernández, Y.: Modelo de Comportamiento Afectivo para Sistemas Tutores Inteligentes (Affective Behavior Model for Intelligent Tutoring Systems), Doctoral Dissertation, (2008)
4. Hernández, Y., Sucar, L.E., Conati, C.: Incorporating an Affective Behavior Model into an Educational Game. In: Lane, H. Ch., Guesgen H. W. (eds.), FLAIRS 2009, pp. 448-453. AAAI Press, Florida (2009)
5. Johnson, W.L., Rickel, J.W., Lester, J.C.: Animated Pedagogical Agents: Face-to-Face Interaction. Interactive Learning Environment, International Journal of Artificial Intelligence in Education, 11, 47-78 (2000)
6. Manske, M., Conati, C.: Modelling Learning in an Educational Game, In Looi, Ch., McCalla, G., Bredeweg, B., Breuker, J. (eds.), AIED 2005, July 19-23, pp. 411-418, IO Press, Amsterdam, The Netherlands (2005)

7. Microsoft Corporation (2005). Microsoft Agent.
http://www.microsoft.com/msagent/default.asp. Accessed November 1, 2005.
8. Murray, R.C., VanLehn, K.: DT Tutor: A Decision-theoretic, Dynamic Approach for Optimal Selection of Tutorial actions, In: Gauthier, G., Frasson, C., VanLehn, K. (eds.) ITS 2000. LNCS, vol. 1839, pp 153-162, Springer, Heidelberg (2000)
9. Zakharov, K., Mitrovic, A., Johnston, L.: Towards Emotionally-Intelligent Pedagogical Agents. In: Woolf, B.P., Aimeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 19-28. Springer, Heidelberg (2008)

Towards Model-Based User Interface Development of e-Learning Management Systems

Josefina Guerrero-García[1], Juan Manuel González-Calleros[1], Jaime Muñoz-Arteaga[2], Miguel Ángel León-Chávez[1], Carlos Reyes-García[3]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México

[2] Universidad Autónoma de Aguascalientes, Mexico

[3] Instituto Nacional de Óptica y Electrónica, Puebla, Mexico

{jguerrero, juan.gonzalez, mleon}@s.buap.mx

Abstract. The article discusses the need for a methodology to support the development of learning management systems. The Web browser has been the traditional way to access such system but emerging technologies presents new challenges. The problem is not just on the technology to support the learning process but also the lack of design knowledge to assist teachers to create content taking into account both the heterogeneity of the learning content and students. These issues are discussed and a proposal is presented to address them.

Keywords: Learning process, workflow systems, multicultural learning objects, e-Learning, model-based development, user interface.

1 Introduction

The design of a teaching-learning process is a task that all professional education must be taken when planning to develop an activity teacher training: course, seminar, etc.

The learning process links users to domain-specific information sources in collaboration spaces designed for knowledge transfer and knowledge generation. Becoming efficient and stimulating for better and effective learning process using available technology requires a strategy to follow. The strategy must consider at least: design of learning content, design of different ways to present content (textual, graphical or mixed) considering different devices (PDA, mobile phones, laptop), and defining collaboration during learning process.

In the design of a learning process, a problem is to correctly identify the context (student, environment, technology available). As depending on this, the teaching strategy, resources, content, and evaluation criteria, are selected. Once content is

created, it must be advertise to those stakeholders involved, such as: teachers, students, managers, etc. [1].

Learning processes are viewed as a workflow (depiction of tasks during which documents or information is passed for one participant to another according to a set of rules) that is recursively decomposed into tasks, that could be associated to a learning object (LO).

The term e-Learning has been introduced to denote learning with the aid of information and communication technology tools [2]. e-Learning still suffers from several usage difficulties, both objective and subjective, such as: the long tradition of classroom education, possible negative experiences with first generation products, a background of badly organized self-teaching attempts, the lack of the typical interaction and emotional relationships that can be obtained with a frontal lesson. The organized problem represents one of the main reasons of on-line courses drop-out.

Recently there are a growing number of researches [3, 4] that put emotions at the centre in the process of teaching-learning. These studies reveal the importance of the learner's emotional states and, in particular, the relationship between emotions and effective learning [4]. However, the influence of emotions in learning processes is not considered in this work.

The obejctive of this work is to define the requierements and to design a possible solution to the aformentioned issues. The organization of this paper is as follows: Section 2 presents a brief background, Section 3 presents our methodology to support learning processes and multicultural learning objects support. Finally, the paper is wrapped up by summarizing our work, deriving conclusions and addressing future trends.

2 Related Work

There is a plethora of computer-assisted e-Learning Management Systems with common elements, such as: tools for creating course material, assessment as well as collaborative tools (forums, emails and chats). These tools achieve the main goal of a system which is to deliver learning content during and after the lecture, i.e. synchronous and asynchronous learning modes.

User Interface Development Methods for e-Learning Environments are scarce, we are not aware of others than [6-9]. In general, a method for designing and developing a learning management system uses a formal specification technique to model the evolution of learning process. In the literature, some methods have been reported and are summarized in Table 1.

The characteristics of the comparison are those that we identified as challanges for a novel e-Learning system, including the design knowledge that is found on formal methods, framework, adaptation or personalization to the user, support to render the User Interface on multiple devices, means to trace learning objectives.

3 Challenges to Create a Methodology to Support Learning Process Definition

Our target is to ensure the transfer of a collaborative learning environment where the user interface (UI) is multi-platform (PC, laptop, and mobile devices) and adaptable to multiple contexts of use (user, device, environment). And in such context facilitate the user (teacher and students primary) the exchange of information more naturally through a UI conceived systematically using this approach.

Table 1. Comparison of collaborative learning environment design methods (Source [6]).

Criteria/Work	(Jonassen et al.) [7]	(McDonald et al.) [8]	(Germán et al.) [9]	(González et al.) [6]
Formal specification technique	*Activity theory*	*Conceptual framework*	*State machine*	*Workflow*
Framework	*NonA*	*C-Flow*	*Cated*	*Ecool*
Personalization	+	+	-	++
Multiple User interface.	--	+	--	++
Traceability of collaborative learning	–	--	+	++

(++ fully supported, + supported, - partially supported, -- not supported)

We argue that creating learning content is an activity that would benefit from the application of a development methodology [10] which is typically composed of:

- A set of models defined according to an ontology. The term "ontology" generates some controversy. It has its history in philosophy, where it refers to the subject of existence. It is also often confused with epistemology, which is about knowledge and knowing. In the context of this research is assumed a set of descriptions of the concepts and relationships within a field of knowledge (learning process).
- A language that expresses these models. In order to specify different aspects and related models, a specification language is needed for allowing designers and developers to exchange, communicate, and share fragments of specifications and that enables the tools to operate on these specifications. These models are uniformly and univocally expressed according to a single Specification Language. A User Interface Description Language (UIDL) is needed and its selection could be based on [11]. A genuine UIDL must be strongly defined based on a trilogy (semantics, syntax, stylistics) [6]. Offering a XML language does not necessarily assures to rely in this trilogy [10].
- Principle-based method manipulating these models based on guidelines. The goal is not to come up with yet another Software Development Method but to reuse existing work and structure it accordingly. The result is a method that structures the development life cycle of learning content in a principle-based way. The method

should promote an exploratory approach having as goal to show a variety of possibilities to encourage design.

— Tools: A suite of software engineering tools that supports the designer and the developer during the development life cycle according to the method. The set of software tools required to support the development of learning content includes:

  - Model editors to assist a designer in constructing the models. These tools consist in syntax editors, form based tools, or visual builders. Some model editors maintain a textual specification consistent with a graphical representation.
  - Design critics provide a designer with quality assessment facilities. Models capturing explicit properties of the artefact are an ideal representation to perform evaluation.
  - Implementation tools translate a specification into a representation that can be used by a compiler, an interpreter or an interface builder.
  - Transformation tools provide support to the designer to edit, store, and execute model transformation rules.

Building the application using the right tools is a trade-off between six main criteria [12]:

1. Part of the application built using the tool. Some tools only support building the presentation part of the application; others also help with low-level interaction, and some support general programming mechanism usable in other parts of the application as well.
2. Learning time. The learning time of the tools varies.
3. Building time. The time required to build a UI using the tool varies.
4. Methodology imposed or advised. Some tools strongly impose a methodology for building the application, such as building the visual part first and connecting it into the reminder of the application afterwards, whereas other tools are more flexible.
5. Communication with other subsystems. Applications frequently use databases, files located on the Web, or other resources that, when supported by the building tool, simplify the development.
6. Extensibility and modularity. Applications evolve, and the new applications may want to reuse parts of existing applications. Supporting the evolution and the reuse of software remains a challenge. Level-4 tools and application frameworks, including Model-Driven Architecture (MDA), inherently promote good software organization, but the others usually lead to poor extensibility and modularity.

The proposed system (depicted in Fig. 1) is composed of several subsystems interconnected together to form an entire collaborative learning environment with social networks. These subsystems will allow on one hand focusing the content and online courses; and on the other hand, students may also use these environments to collaborate with other students in online communities and have advice from teachers college. The following subsections describe more details about the different layers of the methodology.

3.1 Supporting Multicultural Learning Objects

A learning object (LO) is defined as a self-standing, reusable, discrete piece of content broken down into smaller chunks that can be reused in any environment in order to meet an instructional objective. The way LOs are conveyed includes: Web pages, PDF documents, video and/or audio content, animations, and virtual reality to mention a few. LOs have been developed to support virtual learning using technology and pedagogical support. These products can be used under any condition or circumstance where the training or the distribution of the knowledge is required: classroom lessons, staff training in the industry, self-learning process, among others. Adding the multicultural characteristic going beyond regions is a real challenge. For this purpose a LO model is needed. The structure of a LO is specified with: a name, a context, authors name, date, brief description, participants involved, pre and post conditions, and normal and/or alternative learning process flows. The LO model could be a simplified version of SCORM standard. A LO can be part or be composed of other LOs. Also, it can be associated to exercises and/or assessments. The LO is part of a task and will be used in a task, this information is relevant when further we explain how a LO is mapped to a UI from a task model.

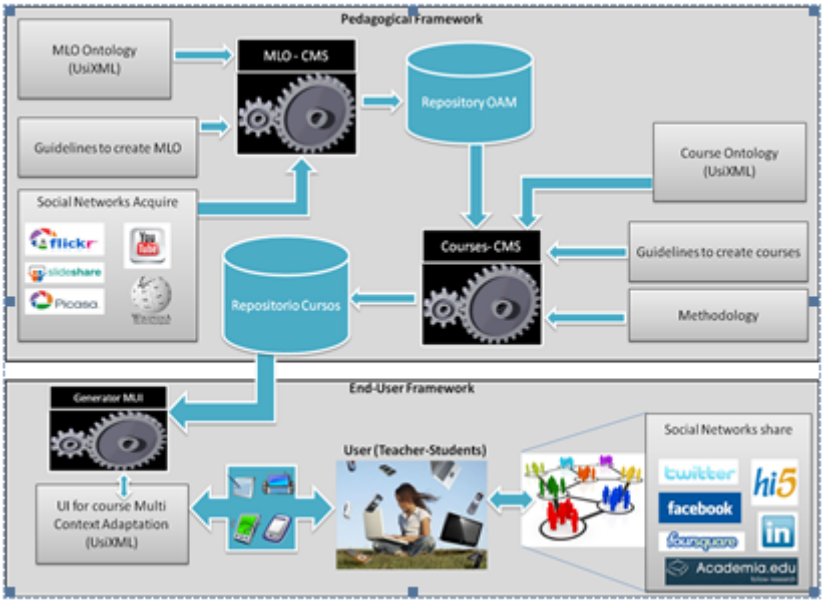

Fig. 1. Methodology to support the learning process definition.

To create the knowledge base related to the Multicultural Learning Objects (MLO) in a formal expression is necessary to make trades to support operations such as extraction of knowledge or recommendations for use, among others. After this, it is

necessary to establish a method that indicates the steps to be followed for the creation of MLO.

Including guidelines to support the method is desirable for the systematic creation of MLO. These guidelines will be based on existing knowledge related to the creation of MLO. Determine the level of automation of the guidelines and application will use the software tool developed for the creation of MLO. This activity focuses on the definition and establishment of methodological guidelines for the creation of MLO. Then, each guideline is classified according to the automation criteria (can be automated or not, or can be partially automated). As the ultimate goal is to automate this method in a software tool, guidelines that cannot be automated at all are useless.

A software tool is desirable for the faster and reusable creation of MLO. It aims to create a Content Management Systems (CMS) for creating MLO. A content manager is a software tool for creating online content simply and massive. This tool will support the proposed method and its use to allow many users to create MLO. One of the requirements for a CMS for MLO creation is the integration of multimedia, such as: audio, video, power point presentation, documents. With the introduction of cloud computing and the use of social networks, such as: slideshare (slides), Wikipedia (free encyclopedia), YouTube (videos), Picasa (photos), among other, to share content, we can avoid storing multimedia in the LO´s repositories. The integration of access to social networks is essential to give versatility to the CMS. Each time you create a MLO, the CMS guides the user with an assistant, and the wizard must go step by step guiding the creation of MLO, relying of the automated guidelines. Special care needs the UI for this MLO-CMS preserving ergonomics, guidelines, heuristics and usability principle. The editor will safeguard, update, edit, access, search, and display settings under different criteria and the MLO repository.

3.2 Supporting Learning Process Definition

A number of online services assist the task of structuring academic courses, relevant and adaptable [13] to the context of students, including learning styles recognition [14]. However, integrating those efforts and to connect them to MLO is more than just a technological problem. A Learning Management Systems (LMS) must consider not just the content adaptation and student learning styles identification, but also to provide teachers means to create content accurately. Most related work assumes that learning content is already there but do not assist the teacher who normally is not an education expert.

For this reason, a mechanism to assist teachers for creating a course is our first requirement for a learning process definition software tool. The software tool should have a CMS for courses. This activity focuses on the development of software tool that serves as content manager for the creation of courses, a module in a learning management systems platform. This tool benefits from the MLO module as it uses material available in the MLO format. The main features for such system are: a) integrating pedagogical recommendations to create a system of guidelines for the creation of courses; b) identify multicultural issues in education; c) identify different

forms of education (classroom, mixed, distance); d) integrate this information in the specification of a learning process; e) integrating intelligent management of information in the learning process; f) integrate MLO information.

As automatic integration of guidelines to use the tool for defining courses content, assistive interaction is needed (wizard, intelligent agent) to guide teachers in this activity. Education experts are needed to define methodological guidelines for the creation of courses. These guidelines should include multicultural aspects in education, different education and different learning paths.

The manager must have a content editor for courses. A learning process can be described as a workflow model [15] that is composed of tasks, resources and places where education takes place. The workflow model is recursively decomposed into learning processes which are in turn decomposed into tasks. Now, as there are many different learning types and approaches to learn, this is believed to occur as a progressive series of tasks, i.e. a workflow. So, a workflow model can be used to plan and to design the process of all aspects of learning. There is a teaching process for the trainers, a learning process for the learners, an organizational workflow for all participants, and a management workflow. All these components interact with each other to form an overall learning workflow. A graphical representation (Fig. 2) of a learning process uses a Petri Net [16] for the specification of processes, big rectangles denoting the rooms where the teaching activity takes place, and roles (students and professors).

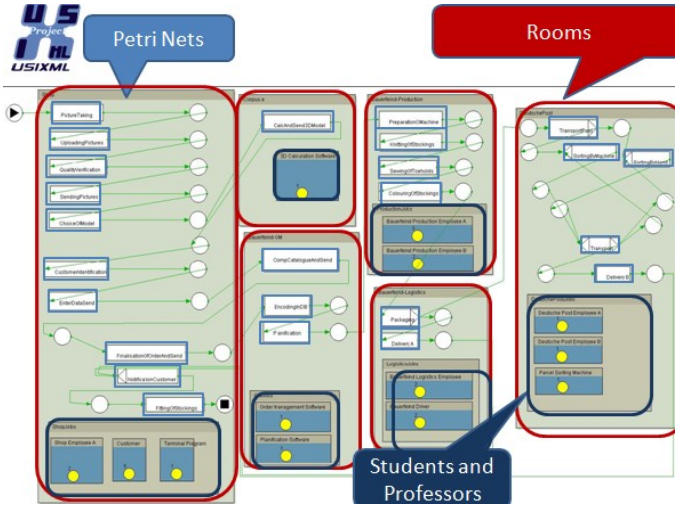

Fig. 2. Learning process editor.

It is vital that the editor considers computer human interface usability. Ergonomic guidelines should be taken into account in the development of this editor; its good design will allow easy use. The editor will safeguard, update, edit, access, search, and display settings under different criteria and the courses. The IU "friendly" is vital and as such must be designed carefully since it depends on the successful use of the

manager. As each rectangle in a Petri net denotes a high level activity or task, more details are needed to describe how those activities are required, in which order they must be executed, thus a task model. There are several notations for task modeling but CTT [17] is a good option because isan expressive and flexible notation able to represent concurrent and interactive activities, also with the possibility to support cooperations among multiple users and possible interruptions. A second reason, task modeling has been used to generate multi-context UI, as it is explained in the next subsection.

This method should also consider elements such as academic monitoring, assessments (for instance using the method of assessment adaptation proposed in [18]), practices and exercises, and other traditional elements considered in a course. The systematic creation of courses based on a method will allow having more quality content. Furthermore, the method will provide the basis to create a tool to automate and thus to create content as a whole.

One important aspect is to consider the other side of the problem, i.e., to consider the student perspective. Showing progression in learning paths progress has been reported [6] as a way to improve users' awareness of its progress; however they fail to motivate him into achieving its goal. How to keep students attach to their learning activities? One idea is to use the famous metaphor created by the site LinkedIn of small progress with great results (your CV online). Seducing the student with the UI is more than rather than the technology itself. Seductive interaction [19] is a way to keep users attached to the system and performing their goals (Fig. 3).

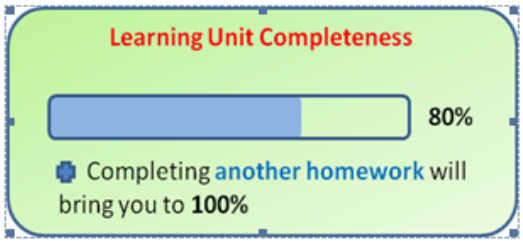

Fig. 3. Learning Progress metaphor inspired from LinkedIn (www.linkedin.com).

3.3 Multi-Context User Interfaces Rendering

Considering that learning process is a description of the foreseen interaction among the following actors: instructors, learners and the system, it captures the context in which the learning process occurs, the process and associated data, and the order of tasks execution; our proposition starts by defining it in a formal definition using a learning process, i.e., a set of tasks. In this step we can indicate the resources involved, the rooms where they work, how and in which order tasks will be executed (using task model [17]); after, MLOs which are associated and stored in a repository, this could be done using the recommended methodology described above or by relying on existing work, for instance MACOBA [20].

Even that the platform of choice for most of the learning environments is the Web browser [21]. A learning portal should seek not only to reuse the tools available on the Web, also will seek to adapt the system to follow the principles of multi-context adaptation of user interfaces (users, environment and platform). Model-Based Development of User Interfaces has been widely reported in the literature [17], [22], [10], to address this problem. The Cameleon Reference Framework [22], the de facto standard, in a simplified description, structures four development steps: 1) Task & Concepts (T&C): describe the various user's tasks to be carried out and the domain-oriented concepts as they are required by these tasks to be performed. 2) Abstract UI (AUI): defines abstract containers and individual components, two forms of Abstract Interaction Objects by grouping subtasks according to various criteria, a navigation scheme between the containers and selects abstract individual component for each concept so that they are independent of any modality.

An AUI is considered as an abstraction of a Concrete User Interface with respect to interaction modality. At this level, the UI mainly consists of input/output definitions, along with actions that need to be performed on this information. 3) Concrete UI (CUI): concretizes an abstract UI for a given context of use into Concrete Interaction Objects (CIOs) so as to define widgets layout and interface navigation. It abstracts a final UI into a UI definition that is independent of any computing platform. Although a CUI makes explicit the Look & Feel of a final UI, it is still a mock-up that runs only within a particular environment. A CUI can also be considered as a reification of an AUI at the upper level and an abstraction of the final UI with respect to the platform. 4) Final UI (FUI): is the operational UI, i.e. any UI running on a particular computing platform either by interpretation or by execution.

The user interface design processes starts with a task model that is processed through an incremental approach to the final UI (Fig. 4 shows the four levels that are involved in the design of a UI using the Cameleon Framework).

From a task model specification it is possible to derive as many UIs as devices have been specified in the framework, for instance UsiXML is capable of rendering learning content on Mobile devices and a Smartphone, as it covers the multi device and multiplatform support. For the complete definition of the method, the reader should refer to [10].

Models are everywhere and are needed to support the Model-Based Development of User Interfaces for learning management system. Even more there is another advantage adopting this approach that is the automatic evaluation and assistance, a key added value if we want to keep the users satisfied.

As reported in [23] the traditional shortcomings of automatic evaluation of UI are addressed by relaying on working with models. The common major shortcoming of any evaluation tool is that the evaluation logic is hard coded in the evaluation engine [23]; for example, two leaders of the web evaluation market, Bobby and A-Prompt only provide the choice of the guidelines set to evaluate: W3C or Section 508, which makes them very inflexible for any modification of the evaluation logic or any introduction of new guidelines. The global process for automatic evaluation of the Model-based approach is depicted in Fig. 5. The "Knowledge Base" contains a formalization of rules for good ergonomics and accessibility. This knowledge base is

a collection from ergonomic guidelines, for instance, structures or various recommendations that are encoded in a formal format, using the UsiXML language.

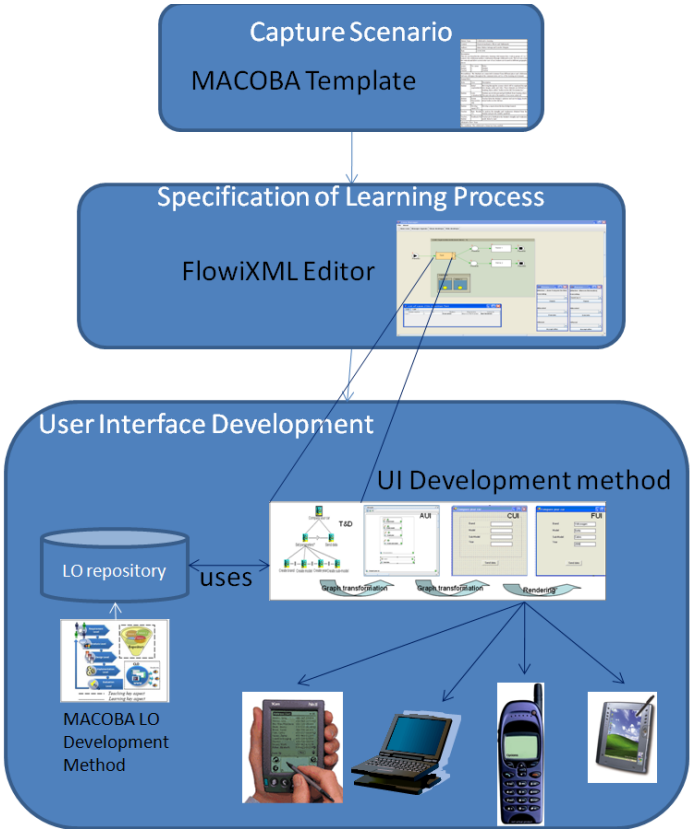

Fig. 4. A Method to generate User Interfaces for a Learning Management System (Source: [23]).

The knowledge base is used by the "Formal rules compiler" to load and parse the rules. Once this internal structure is created the tool performs a data analysis of the UI, encoded in UsiXML, which may be developed in a UsiXML editor. The UsabilityAdviser search for violations of rules formalized through the automatic evaluation of UI data. Finally, a report of the found violations of ergonomics and accessibility is presented. One major challenge is to create and update the knowledge base on ergonomic rules, which requests a complete review and compilation of existing rules from different sources. These rules are often expressed in a natural language that is normally more complex and open compared to a programming language. Anyway, the UsabilityAdviser provides an extensible way of evaluation from multiple sources of guidelines for (parts of) a UI.

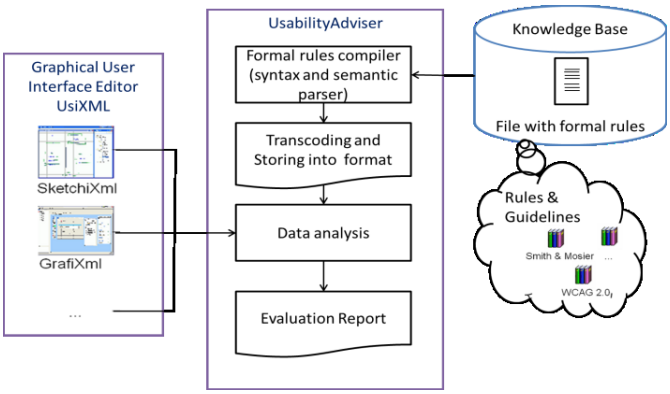

Fig. 5. Global process for automatic evaluation.

4 Conclusions

This paper presents a formal methodology to design an e-Learning for multicultural learning objects. A learning process is described as a workflow model that is composed of tasks, resources and places where education takes place. The goal of using the methodology is to provide the design knowledge to develop an eLearning Management Systems that addresses the current needs of e-Learning systems, including: gamification, multicultural, social networking, formal definition of learning process, multi-user support, multi-environment use, and multi-device. We propose to rely on a model-based engineering approach for at least two reasons: first, it enables structuring the development life cycle of learning process in a principle-based way (guidance support); second, the final rendering of UI for multi-context environments is desirable.Ergonomic guidelines are taken into account in the proposed solution, since its good design will allow for easy use. The future of this work is to develop the proposed solution.

References

1. Hilera, J. R., Palomar, D.: Modelado de procesos de enseñanza- aprendizaje reutilizables con XML, UML e IMS-LD. RED, Revista de Educación a Distancia, número monográfico III. Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables (II), 58-69 (2005)
2. Georgiev, G., Dimitrova, T., Karamanska, D.: Ergonomic Factors of Computer-Assisted Learning, VMEI Sofiam– in Bulgarian (1988)
3. Currin, L.: Feelin'groovy: Experts now believe that e-learning must elicit positive emotions to succeed. http://elearnmag.acm.org/archive.cfm?aid=968474
4. Kort B., Reilly R., Picard R.W.: An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy-Building a Learning Companion, In: Toshio Okamoto, Roger Hartley, Kinshuk, John P. Klus (eds.) Proceedings of International

Conference on Advanced Learning Technologies (ICALT 2001), pp. 43-48. Madison Wisconsin.(2001)
5. Fardoun, H., Montero, F., López-Jaquero, V.: eLearniXML: Towards a model-based approach for the development of e-Learning systems considering quality. Advances in Engineering Software, 40 (12), pp. 1297-1305. (2009)
6. González, J.M., Guerrero, J., Muñoz, J., Vanderdonckt, J., Martinez, J.: A Method for Generating Multiplatform User Interfaces for E-Learning Environments, In: T.-T. Goh (ed.), Multiplatform E-Learning Systems and Technologies: Mobile Devices for Ubiquitous ICT-Based Education, pp. 90-111. IGI Global Inc., Hershey (2009)
7. Jonassen D.J., Rohrer-Murphy, L.: Activity Theory as a framework for Designing Constructivist Learning Environments, Educational Technology Research and Development. 47 (1), pp 61-79. (1999)
8. MacDonald, C.J., Stodel, E.J., Thompson, T-L., Muirhead, W., Hinton, C., Carson, B., Banit, E.: Addressing the eLearning Contradiction: A Collaborative Approach for Developing a Conceptual Framework Learning Object. Interdisciplinary Journal of knowledge and Learning Objects. 2, pp. 79-98 (2005)
9. Germán-Sánchez,V., Téllez-Mora, F. & Morales- Gamboa, R.: Collaborative Environment for the Composition of Learning Scenarios. In: Gelbukh and Zechinelli (eds.) Proceedings of ENC07 Conference, Mexico. IEEE Editorial. (2007)
10. Vanderdonckt, J.: A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In O. Pastor & J. Falcão e Cunha (eds.) CAiSE'05 (Porto, 13-17 June 2005). LNCS, vol. 3520, pp. 16-31 Springer-Verlag, Berlin (2005)
11. Guerrero-García, J., González-Calleros, J.M., Vanderdonckt, J., and Muñoz-Arteaga, J.: A Theoretical Survey of User Interface Description Languages: Preliminary Results. In: Proceedings of LA-Web/CLIHC'2009 (Merida, November 9-11, 2009), pp. 36-43, IEEE Computer Society Press, Los Alamitos, (2009)
12. Shneiderman, B., Plaisant, C.: Designing the User Interface. 4th Edition, Addison Wesley, Reading (2204)
13. Canales-Cruz, A., Peredo-Valderrama, R.: Adaptive and Intelligent Agents Applied in the Taking of Decisions Inside of a Web-Based Education System. In: Ngoc Thanh Nguyen, Lakhmi C. Jain (eds.) Intelligent Agents in the Evolution of Web and Applications.167, pp. 87-112. (2009)
14. Zatarain-Cabada, R., Barrón-Estrada, M.L., Ponce-Angulo, V., García, A., Reyes-Garcia, C.: A Learning Social Network with Recognition of Learning Styles Using Neural Networks. In: Martinez-Trinidad, José Francisco; Carrasco-Ochoa, Jesús Ariel; Kittler, Josef (eds.) Proceedings of Second Mexican Conference on Pattern Recognition MCPR. LNCS, vol. 6256, pp. 199-209, Springer Berlin Heidelberg (2010)
15. Guerrero, J., Vanderdonckt, J. & Gonzalez, J. M.: FlowiXML: a Step towards Designing Workflow Management Systems. Journal of Web Engineering. 4, 2, pp. 163–182. (2008)
16. Aalst, W., Hee, K.: Workflow Management: Models, Methods, and Systems. The MIT Press, Cambridge (2004)
17. Paternò, F.: Model-based design and evaluation of interactive applications. Applied Computing. Springer, Heidelberg (2000)
18. Barbosa, H., García-Peñalvo, F., Rodriguez-Conde, M.: Use of the Question and Test Specification to Define Adaptive Test. In: Knowledge Management, Information Systems, E-Learning, and Sustainability Research, Communications in Computer and Information Science. Vol. 111, pp. 13-21. Springer Berlin Heidelberg (2010)
19. Anderson, S.: Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences, Poet Painter, USA (2011)

20. Margain, L., Muñoz, J., Álvarez, F.J.: A Methodology for Design Collaborative Learning Objects. In: Proceeding of 8th IEEE International Conference on Advanced Learning Technologies, pp. 87-91.IEEE Press, Santander, Spain (2008)
21. Bär, H., Häussge, G., Rößling, G.: An integrated system for interaction support in lectures. In: Proceeding of the 12th Annual SIGCSE conference on innovation and technology in computer science education ITiCSE '07, pp. 281- 285. ACM, New York, NY(2007)
22. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers. 15(3), 289–308 (2003)
23. Osterloh, J., Feil, R., Lüdtke, A., Gonzalez-Calleros, J.: Automated UI Evaluation based on a Cognitive Architecture and UsiXML, Software Support for User Interface Description Language. In: Coyete, A., Faure, D., González, J., Vanderdonckt, J. (eds.) Proceedings of the INTERACT Workshop Software Support for User Interface Description Language (UIDL), pp. 237-241.Thales, France (2011)
24. Vanderdonckt, J., Beirekdar, A.: Automated Web Evaluation by Guideline Review, Journal of Web Engineering, 4 (2), pp. 102-117 (2005)

Fermat: An Intelligent Social Network for Mathematics

María Lucía Barrón-Estrada, Ramón Zatarain-Cabada, Rosalío Zatarain-Cabada,
Jesús Armando Beltrán Verdugo, Franceli Linney Cibrian Robles,
and Marsia Irais Quiroz López

Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe,
Culiacán Sinaloa, 80220, Mexico

{lbarron, rzatarain}@itculiacan.edu.mx,
{armando.3eltran, Linney11}@gmail.com, marcirais@hotmail.com

Abstract. We present Fermat, an Intelligent Social Network for Mathematics Learning, which integrates an Intelligent Tutoring System as an extra feature to help improve the teaching and learning process. The intelligent tutor takes into account both cognitive and affective factors, and by use of artificial intelligence techniques, provides users with a personalized and satisfying experience when taking the courses.

Keywords: Intelligent Tutoring Systems, social networks, neural networks, emotion recognition.

1 Introduction

Social Networks (SN) have a great impact on the daily lives of many people. In [7] the authors describe it as a structure of nodes that represent individual relationships (or organizations) among people of a certain domain. So, we can say that a social network is a dynamic interaction which allows sharing different types of files, comments and topics [3].

There are several types of social networks, which are distinguished according to the approach they have. For example, social learning networks, which use a collaborative environment and tools to help users in the learning process. Tools that can integrate social learning networks are Intelligent Tutoring Systems.

Intelligent Tutoring Systems (ITS) are computer programs that use many of resources to support the teaching and learning process. ITS must incorporate techniques of Artificial Intelligence (AI) and education, with the aim of creating a flexible and interactive environment that considers the different cognitive styles of students [11]. ITS should play an important role in monitoring both the learning the student has achieved, as well as identifying weaknesses, in order to find strategies that fit the student's cognitive style [11].

In order to reduce the problem of learning mathematics in Mexico middle schools, we built an intelligent social network for mathematics named Fermat [15]. Fermat contains an Intelligent Tutoring System whose main objective is to support the teaching-learning process in formal classroom education.

This paper is organized as follows. In Section 2 we present a general architecture of Fermat. Section 3 gives information about the analysis and design of the social network. Section 4 explains how the ITS is structured. Section 5 will discuss some results and finally conclusions are presented in Section 6.

2 Fermat Architecture

Learning Social Network **Fermat** has the basic functionalities in all social networks, but its main feature is that it includes an ITS that offers the course content in a personalized style to users, as shown in Figure 1.

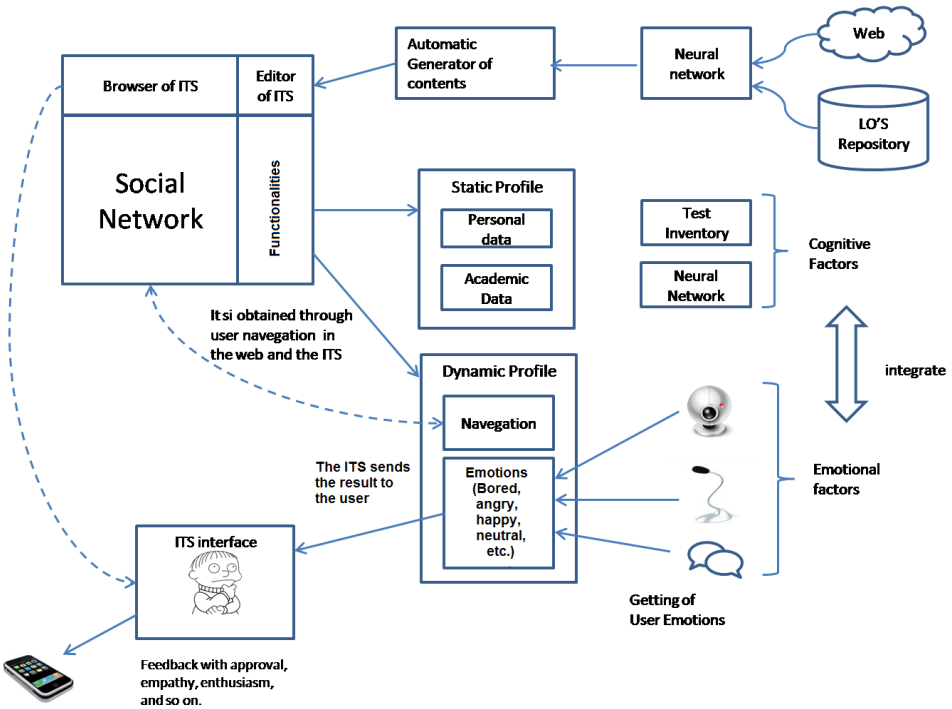

Fig. 1. Fermat Architecture

Users of the network are associated with personal, academic and affective information in a profile, which is obtained statically and dynamically. The static profile contains the initial information of the user (e.g. personal and academic information). The dynamic profile will be updated according to the user interaction within the network and the ITS, taking into account during this interaction, cognitive and emotional aspects.

According to [6], emotions are closely related to student learning, which in our point of view, represents a key factor to the student results.

Cognitive factors are obtained according to the history that we obtain from the results of examinations of the user and the learning style computed by the neural

network. Emotional factors are obtained by sensors that are monitoring the user's emotions through facial expression and voice sound.

3 Analysis and Design of Fermat

In the analysis phase we determined the objectives and user requirements of the social network, which are fundamental to understanding the features needed by the user to interact and use the network.

These features provide a set of functional requirements that the network must have as well as quality requirements. Every functional requirement was developed in a use case to show how the user will interact with the system. We also developed a context diagram to show how the system interacts with external actors.

The design phase of the social network was divided in four main topics: data model, architecture, interfaces and components.

The system architecture was shown in figure 1, and some interfaces are presented in section 5. The components of the social network such as the editor and the browser, the user management, the learning objects, the collaboration of users in the network, and the adaptation of courses to a learning style, are directly related to achieving one of the primary functions of a social network, which is to enhance collaborative learning using an intelligent tutor system.

For the design of the interfaces we used the Mockup Builder tool [8].

4 Fermat ITS

The intelligent tutoring system (ITS) developed for the Fermat social network includes three main components (figure 2), which together are able to determine what the students know and how they are progressing, adjusting to their learning style needs [11].

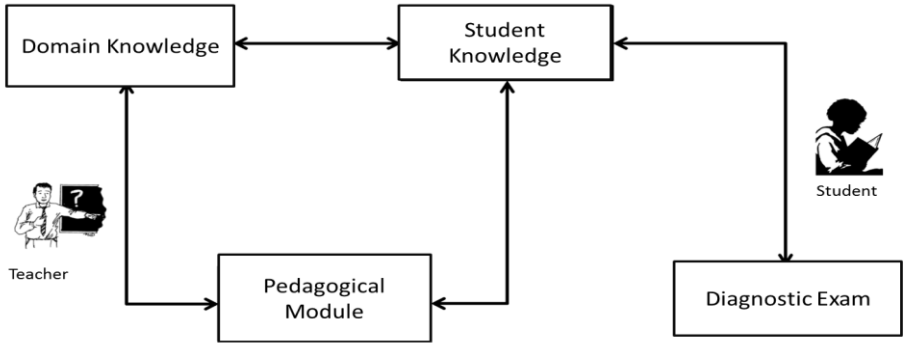

Fig. 2. General Architecture of the ITS.

The three main components are:

1. Expert or Domain Knowledge Module.
2. Student Knowledge Module.
3. Tutoring or Pedagogical Module.

Domain knowledge Module: it is one that contains the description of knowledge on the subject of a particular domain, providing knowledge of what is taught to the student, so he can gain the skills and concepts required for meaningful learning. A course in Fermat can be seen as a tree diagram containing chapters and in turn these are made by subjects, as shown in Figure 3. The totality of all nodes in the tree represents the expert knowledge.

Student Knowledge Module: This module is responsible for assessing the student performance to determine his/her cognitive abilities and reasoning skills. It provides the information about what a student knows. This module is central to the module tutor in the selection of the learning style that better suits the user.
Fermat realizes what the student's knowledge is through a diagnostic test. The test results show what the student knows and what he needs to learn. The Fermat student module can be seen as a sub-tree of all knowledge possessed by the expert in the domain, as shown in the left part of figure 3. The representation is based on a model called "Overlay", where the student's knowledge is a subset of the expert knowledge. As the student uses the intelligent tutor he expands this subset [5].

Pedagogical Module: It represents the fundamental strategies for teaching the course content in Fermat. It is responsible for selecting the appropriate learning style and provides assistance to the student. For example, the tutor must know how to respond when the student cannot answer a question. In Fermat, the tutor module is based on the model developed by Felder and Silverman, which classifies the student preferences on four dimensions: perception, processing, input, and understanding [4].

On the other hand, we integrated another module to recognize the emotional states of students. Emotions are detected by means of the expression of the face and by the voice.

The method used for the detection of visual emotions is based on Ekman's theory [13], which recognizes ten emotions: anger, disgust, fear, happiness, sadness, surprise, well, bored, tired, neutral. To determine the emotion, the system take a picture of the student´s face, sending the image file into a module to be transformed to a more basic form. Based on this picture we get the feature points that minimize the set of input data to the neural network. We use a Kohonen Neural network with 20X20 input neurons and 2 output one is representing the emotion.

For the detection of emotions in the voice, this is captured primarily through the computer microphone and then normalized. Then we apply the technique to characterize components analysis (PCA) to the signal representing the voice. After using the SFFS method [14] we obtain an optimal set of features that will feed the neural network.

In Figure 4, we can see how we integrate the visual and sound emotions using a neural network. Having recognized the emotion, this is sent to the intelligent tutor to respond based on the emotion.

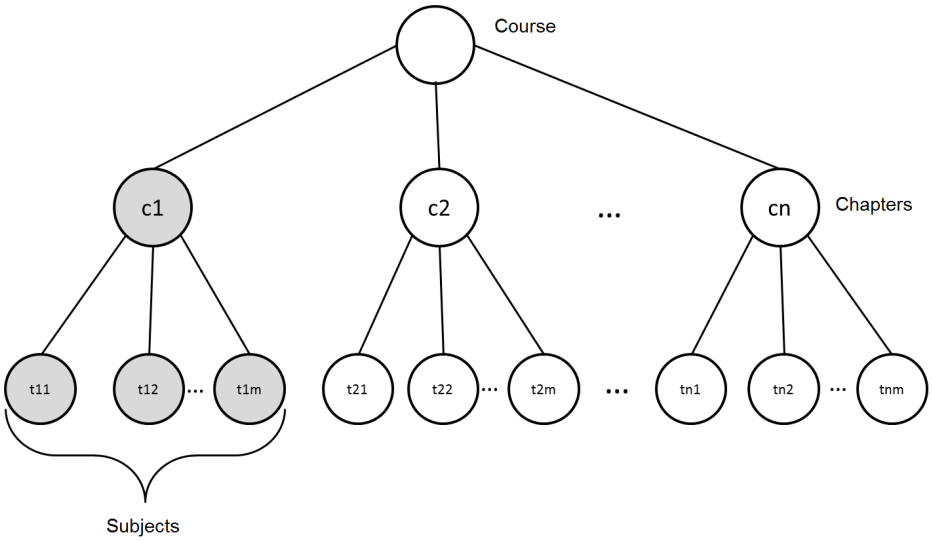

Fig. 3. Basic structure of a Fermat course.

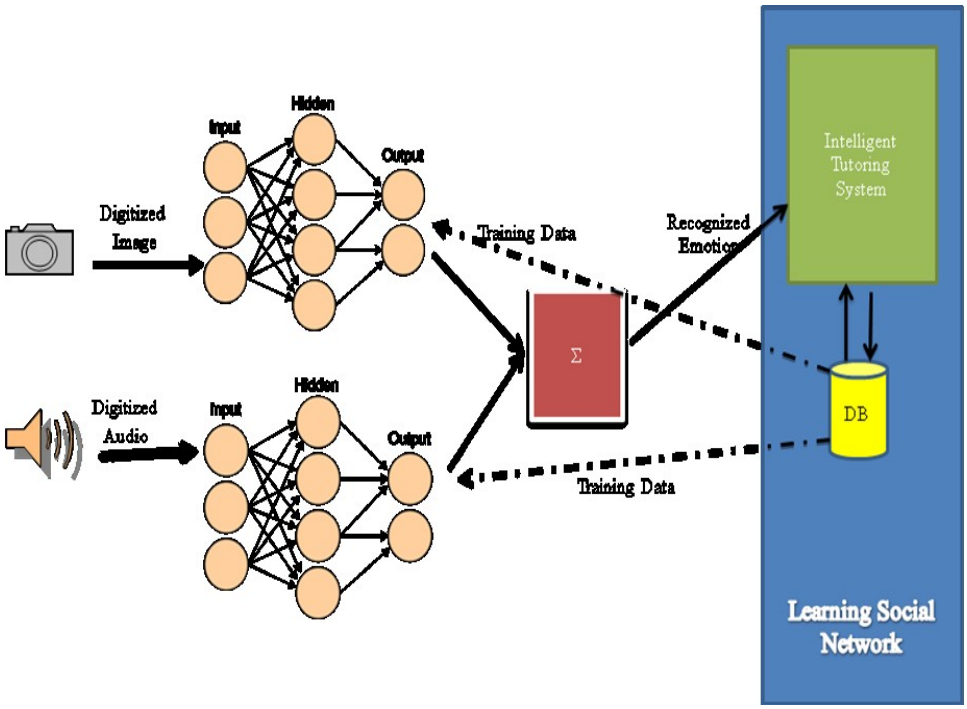

Fig. 4. Emotion recognition**.**

5 Fermat Testing

Fermat was tested with primary school students in third grade. During this period we constantly supervised the use of the social network by the students and the teacher.

First, users tested the common functionality in the network such as: edit your profile, add friends and send messages. A teacher created a learning community, and added his students as members. The teacher also created a course as is shown in figure 5.

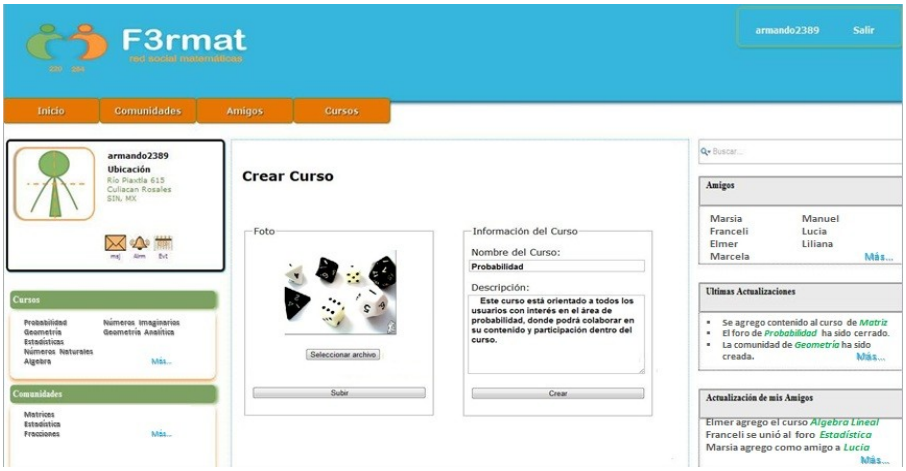

Fig. 5. Creation of a course in Fermat.

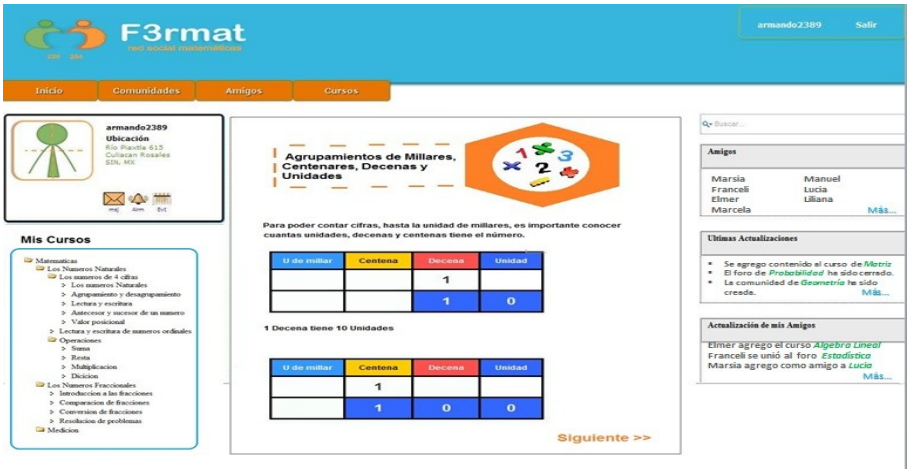

Fig. 6. Displayed Course in Fermat.

Later, the course was browsed by the students (Figure 6).

At the end, students answered tests, which updated their sub-tree of knowledge, and the intelligent tutor showed them new information on the topics where the student presented problems.

6 Conclusions

Social networks are a tool where users can find a meeting space, making promoting a cooperative attitude of cooperation. Adding an Intelligent Tutoring System to a Social Network helped the teacher with the teaching process in order for students to gain meaningful learning. The actual results show that an Intelligent Social Network for Learning Mathematics can assist children to achieve better results in a context of traditional or formal education.

References

1. Aced, C.: Redes Sociales en una semana. Barcelona, PAPF, G. (Ed.) (2000)
2. Boyd, D. & Ellison, N. B.: Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication, 13 (2007)
3. Calvo Muñoz.: Networking, Uso práctico de las redes sociales. Madrid, M. ESIC (Ed.) (2009)
4. Graf, S., Rita Viola, S., & Leo, T.: Representative characteristics Of Felder-Silverman Learning Styles: An Empirical Model. In: Proceedings of CELDA 2006, IADIS Int. Conf. on Cognition and Exploratory Learning in Digital Age. IADIS Press pp. 235-242 (2006)
5. Günel, K.: Intelligent Tutoring Systems: Conceptual Map Modeling. LAMBERT Academic Publishing (2010)
6. Kort, B., Reilly, R., & Picard, R. W.: An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy—Building a Learning Companion. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT '01). IEEE Computer Society, pp.43 (2001)
7. Liccardi, I.; Ounnas, A.: Pau, R.; Massey, E.; Kinnunen, P.; Lewthwaite, S.; Midy, M.-A. & Sarkar, C.: The role of social networks in students' learning experiences SIGCSE Bull., ACM, vol. 39, pp. 224-237 (2007)
8. Mockup Builder @ copyright 2011. http://mockupbuilder.com/App (2011)
9. Patro, B.: Utilización de la Web 2.0 para aplicaciones educativas en la U.N.V.M. Eduvin. México, D.F. (2010)
10. Secretaria de Educación Pública. Enlace boletín informativo. Consultado el: 29 de agosto de 2010] http://enlace.sep.gob.mx/ba/docs/boletin_enlaceba2010.pdf (2010)
11. Stankov, S., Glavinic, V., Rosic M.: Intelligent Tutoring Systems in E-Learning Environment: Design, Implementation and Evaluation. Information Science Reference (2011)
12. Wenger, E. Comunidades de práctica. Aprendizaje, significado e identidad, UOC (2001)
13. Ekman P, Oster,H.: Facial expressions of emotion. Annual Review of Psychology, vol. 30, pp. 527-554 (1979)
14. Pudil P, Novovičová J, Kittler J.: Floating search methods in feature selection. Pattern Recognition Letters, 15(11) (1994)
15. Sean, M.: The Mathematical Carrer of Pierre de FERMAT. Princenton University Press (1994)

Intelligent Tutoring and Training Tools for the Electric Power Sector Developed at IIE

Alberto Reyes, Yasmín Hernández, Pablo de Buen, Eduardo Islas, Miguel Pérez, Carlos F. García–Hernández, Guillermo Rodríguez, Rogelio Martínez, and Fernando Jiménez

{areyes,myhp,debuen,eislas,mperez,cfgarcia,gro,remr,fjimenez}@iie.org.mx
http://www.iie.org.mx

Abstract. The electric power industry requires qualified personnel to support an optimal and safe operation. Since its beginnings, the IIE has been developing different training technologies and systems for CFE, the main utility for generation, transmission and distribution of electric power in Mexico. Some of these endeavors provide tailored instruction considering trainees traits such as learning styles, affective states and current knowledge. Other developments are focused on enabling multi-functionality. The IIE has also developed intelligent assistant systems, virtual reality systems, and power plant simulators. Besides this, the IIE is interested in developing e-learning platforms to support CFE's personnel training. This paper presents a summary of these developments.

Keywords: Adaptive instruction, intelligent assistants, virtual reality systems, training simulators, learning environments.

1 Introduction

The Electrical Research Institute (IIE) is a public institution dedicated to innovation, technological development and applied scientific research, within the electrical and oil industries, and one of its main functions is to support the Federal Electricity Commission's (CFE) processes. These processes are complex and involve different levels of risk. Thus, one of its main concerns is to make sure that its personnel is well trained and qualified in order to prevent accidents and equipment damage, as well as to reduce operational costs. Traditionally, personnel training was achieved by attending specialized courses. However, nowadays the industrial development demands opportune and flexible training. The IIE has developed different tools and training approaches, based on modern technologies, to support these challenges.

This paper describes some of the training tools developed by IIE. First, it presents two works focused on the importance of providing tailored instruction. These consider both learning styles theory and the student affective state. Then, it describes three multi-functionality systems namely: an intelligent assistant in the power plant domain, an adaptive system for learning and consulting engineering procedures, and an assistant system for the design of electrical distribution

substations. After that, it explains the architecture followed by different systems based on virtual reality, one devoted specifically to live line maintenance training in different tensions, and other for the fossil fuel power plant domain. Later on, it talks about a power plant simulator complemented with an expert system for training of power plants operators. Then, it describes an e-learning platform constructed by the IIE. Finally, some conclusions are presented.

2 Strategies for Adaptive Instruction

2.1 Learning Styles for Intelligent Learning Environments

Until now, most of the intelligent learning environments personalize instruction basically by tracking what a student knows. However, adapting instruction is not only concerned with students current knowledge; there are many other proposals to adapt the lessons to other aspects of students like skills mastery, motivational and affective state, self-efficacy, learning styles, among other proposals. We have developed a general framework to adapt the instruction based on the Felder-Silverman Learning Styles Model [1]; the instruction is presented according to a set of rules taking into account the student's learning style which is identified by an assessment instrument given to students [2].

The learning styles theory relies on the hypothesis where each individual has a particular way to learn, including strategies and preferences, emphasizing that individuals perceive and process information in different ways. Consequently, the learning styles theory states that individuals learning has more to do with a process focusing on the learning style than with the individuals intelligence.

The Felder-Silverman categorizations of learning styles are: Active and reflective learners. The active learner better understands information by doing something with it and it likes group work. The reflective learner understands information better by thinking about it quietly first and it prefers to work alone. Other categorizations consider the existance of sensing and intuitive learners, visual and verbal learners, and sequential and global learners.

The sensing learner likes learning facts and solving problems by using well-established methods, but it dislikes complications. The intuitive learner prefers discovering possibilities and relationships and likes innovation but dislikes repetition. The visual learner remembers better what he/she sees: pictures, diagrams, flow charts, time lines, films, or demonstrations. The verbal one gets more from words and written and spoken explanations. The sequential gains understanding in linear steps and follows logical stepwise paths in finding solutions. The global one learns in large jumps and solves complex problems quickly once they have grasped the big picture.

To identify the learning style of a person, we use the Felder-Silverman assessment instrument, which is a Soloman and Felder questionnaire with 44 questions [3]. The collection of rules proposes a set of teaching instructions for each learning style [4]. Table 1 shows rules for active/reflective learners.

To apply these rules, every lesson of a course has to be converted into 8 different lessons according to the teaching instructions. This effort is justified

Table 1. Rules of teaching instructions for active and reflective learning styles

Learning style	Teaching instructions
Active	Show exercises at the beginning of the chapter because they like challenges and problem solving Show less examples. They are not interested in the way others have done something, because they want to solve a problem by themselves
Reflective	Show exercises at the end of a chapter Show examples after explanation content, but before exercises Show less exercises, because they learn better by thinking about a topic instead of solving problems actively.

when there are many potential students classified in each of the learning styles so that they can better profit personalized learning objects.

To develop our proposal, we have assembled a site using Moodle with instructional material composed of lessons, tests, and exams in compliance with the SCORM standard (Shareable Content Object Reference Model). Currently, we are basing our proposal on an algorithm rooted in artificial intelligent planning techniques. In this way, an individual course is generated for each student based on his/her individual needs and his/her learning goals.

2.2 A Model of Affective Behavior

Emotions have been recognized as an important component of motivation and learning. There is support that experienced human tutors look at and react to the emotional state of students in order to motivate them and improve their learning process. In the few past years, there has been extensive work on modeling student emotions; however, there have been only limited attempts to integrate information on student affect in the tutorial decisions. To tackle this problem, we have developed an Affective Behavior Model (ABM).

The ABM takes affect into account when interacting with a student by inferring the affective state of the student; and by establishing the optimal tutorial action based on the students current affective state (besides knowledge). A diagram of the ABM is presented in Fig. 1. The model is composed of an affective student model and an affective tutor model. The tutor model produces an affective action considering the affective and pedagogical student models as well as the tutorial situation. The affective action is a component of the tutorial action to be presented to the student.

The affective student model is based on the OCC model of emotions [5], and relies on a Bayesian network [6]. The OCC model considers emotions as a result of a cognitive appraisal between situation and individual goals. In our model, goals are inferred based on the Five-Factor personality model [7].

The affective tutor model is designed based on interviews and surveys with 20 qualified teachers aimed at understanding which actions the teachers select according to the state of a students affect and knowledge. We asked them to say what they do when they are teaching. The teachers watched videos of students

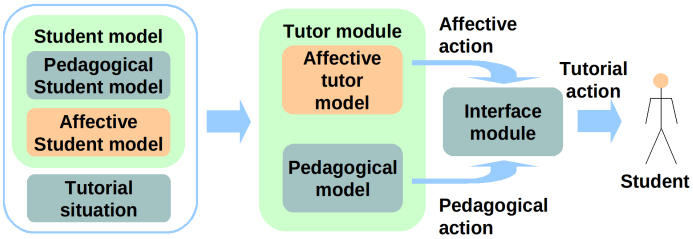

Fig. 1. General diagram for the Affective Behavior Model. The model is composed of an affective student model and an affective tutor model.

and they established the suitable affective and pedagogical actions for each tutorial situation, and they stated why they do those actions. The affective tutor model selects the tutorial actions with the best expected effect on a students affect and knowledge by using a dynamic decision network with a utility measure on both, learning and affect [8].

The ABM was tested in a controlled user study and in a wizard-of-oz study; 82 students participated. In both studies the tutorial action was delivered by an animated agent. The results of these studies are encouraging, since they show a high precision in the affective student model comparing with self-reports and they show positive impact on students learning comparing pre-tests with post-tests.

3 Multi–functional Intelligent Assistants

3.1 ASISTO: An Intelligent Assistant System for Power Plant Operators Training

ASISTO (standing for *Operation ASSISTant* in Spanish) [9] is an intelligent assistant system composed of a training module that includes an instructor console conected to a plant simulator. The main component of the training mode is the explanation system which is aimed to provide new operators with background information during a training session. The automatic generation explanation mechanism is composed of two main stages. In the first stage, the most *relevant variable* is obtained by analyzing a Markov decision process (MDP)[10] used by the operation assistant. This relevant variable is defined as the factor that has the greatest impact on the utility given certain plant state and recommendation, and it represents a key element in the explanation generation mechanism. In the second stage, an explanation is generated by combining the information obtained from the MDP analysis, and displayed in the form of a general template. The current state, the recommended action generated by the MDP, and the resulting relevant variable are then used as pointers to query a domain knowledge base

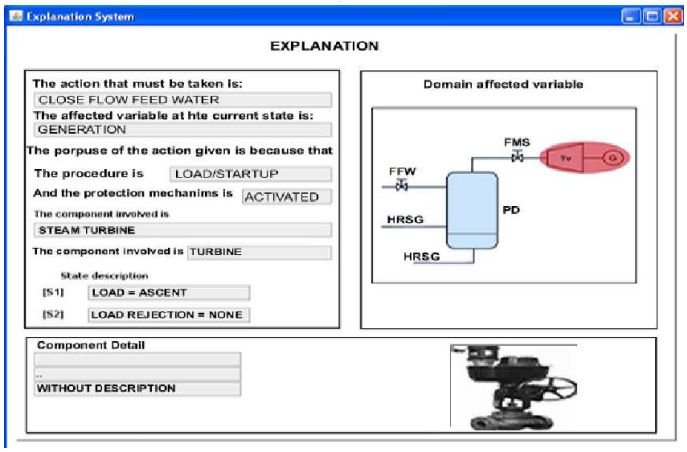

Fig. 2. This template is composed of 3 main parts: (i) the recommended action and the relevant variable in the current situation; (ii) a graphical representation of the process highlighting the relevant variable, and (iii) a verbal explanation.

and extract the relevant information to fill–in the explanation template. A more detailed description of the explanation mecanism can be found in [11].

ASISTO generates an explanation for every user level: novice, intermediate and advanced. Advanced users do not require a well detailed explanation so that they are provided with essencial information only. However, novices might need more specific explanations and a template with more complete information might be displayed. Fig. 2 shows an example of explanation templates for advanced users. In general, the template shows the optimal action on the left hand side, explaining why it is important to perform the recommended action, the component associated to the optimal action, and a brief description of the current plant state. On the right side of the relevant variable there is a diagram of the process associated with the action executed by the user.

When running experiments, operators trainees experienced improvements in their general performance after using the explanation system. We plan to conduct additional user study tests using the explanation module in order to demonstrate that the quality of explanations generated automatically is of a very high standard when compared against those given by a domain expert.

3.2 Multi-functional Knowledge Based System to Learn, Apply and Consult Procedures

Lacepro is an adaptive multi-functional knowledge based system for Learning, Applying and Consulting Engineering Procedures [12]. The system achieves its multi-functionality by using a single knowledge representation scheme that facilitates the tutoring, problem solving and consulting tasks, the representation of a user's model and the automatic generation of examples and evaluation prob-

lems. As shown in Fig. 3, the representation of the domain is exploited by three different knowledge operators (the "Tutor", the "Consultor" and the "Problem Solver"), each of which is in charge of the tasks of tutoring, consulting, and problem solving, respectively. The three knowledge operators consult a user model in order to tailor their explanations to what the user knows and prefers [13]. LacePro uses a machine learning mechanism to learn the rules that result in "good" strategies that are effective for different classes of users [14]. This mechanism is also used to adapt these rules to the preferences of particular users.

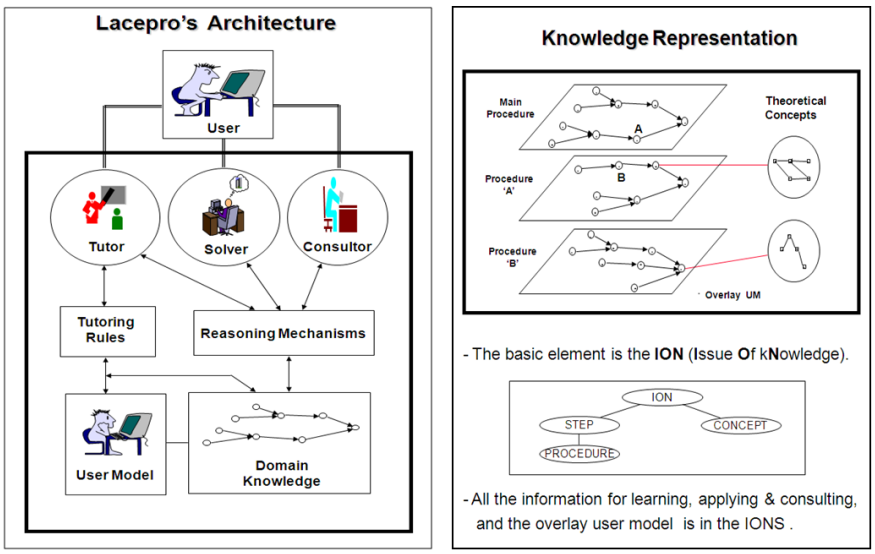

Fig. 3. Lacepro's Architecture and Knowledge Representation. They are the basis for multi-functionality and adaptability to the userss knowledge level and preferences.

In Lacepro, the domain knowledge is represented through a set of goal networks (see Fig. 3). Each node of these networks represents a step within a procedure. A step is a subgoal within a procedure (i.e., the calculation or selection of a parameter or set of parameters through: direct assignment using formulae, production rules, data retrieval from tables, or specialized routines). The links among nodes represent the flow of the procedure. Those concepts required to understand the basis of a step are stored in concept nodes that are linked to the step node.

Lacepro shows that it is possible to develop transparent systems that adapt and evolve continuously as professionals learn and apply procedures as part of their work.

3.3 SiDSED: A System for Designing Electrical Distribution Substations

SiDSED [15] is a system for designing electrical distribution substations that uses different levels of building blocks to simplify the design process and facilitate the estimation of costs of new electrical substations. The building blocks are based on three levels of abstraction; buildings blocks of the highest level are composed from building blocks of lower levels. Each building block has an associated cost obtained from a concepts catalogue with unit prices. The system was developed for power distribution at CFE.

SiDSED was developed in three modules: engineering design module, costs engineering module and visualization module (see Fig. 4).

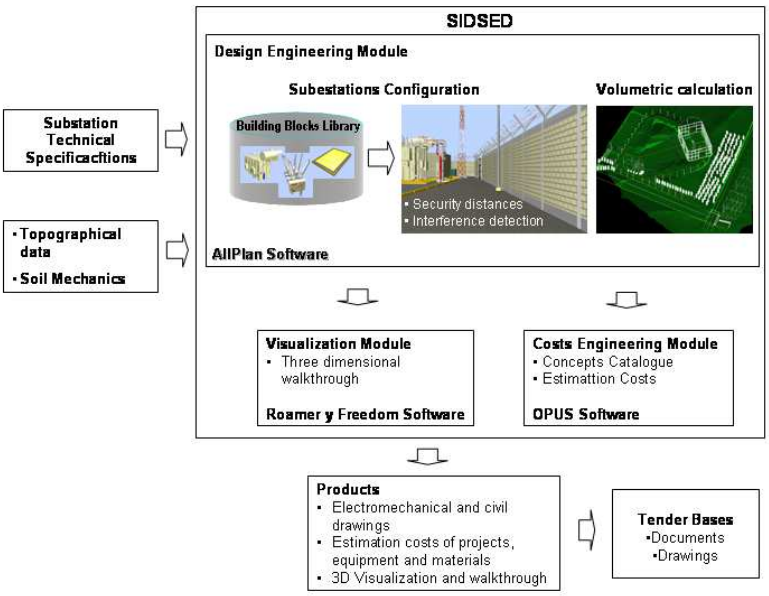

Fig. 4. SIDSED Modules: Engineering design module, costs engineering module and visualization module

In the engineering design module, human designers use the building blocks to design a new electrical substation, taking into account some topographical data. After that, the cost of an electrical substation is estimated with the costs engineering module. Finally, with 3D visualization and a walkthrough module, designers can make decisions about aspects related with construction, operations, maintenance and training.

Some of the benefits obtained with the use of this type of approaches are related mainly to cost savings through design automation, reduction of construction problems and faster throughput of projects. The building blocks can

be 3D standards for advanced engineering, automated drawing, data extraction and reusability of designs. We can use it virtual substations for site selection, community and government acceptance, and 3D visualization and walkthrough can be used to improve construction, commissioning, operations and maintenance. Nowadays, we are training personnel and implementing SIDSED in all CFE's divisions along the country.

4 Virtual Reality Systems

4.1 Virtual Reality Systems for Maintenance Training

In July 2003, the Virtual Reality Group (GRV) was founded at the IIE. Since then most of the GRVs efforts have been devoted to develop training systems for CFE. The first Virtual Reality (VR) System developed was ALEn3D (acronym in Spanish for training system for medium live line maintenance). From then on other different training systems were developed, such as: high tension lines, transmission lines, underground lines, and still in development substation maintenance tests (Fig.5 right). They all share the same architecture defined for ALEn3D, which includes three operation modes, namely: learning, practice and evaluation modes (Fig. 5). All these systems are able to keep track of the students progress. The systems are able to remember who used the system, what maintenance procedure was learned, in what step and when the student was using the system, etc. It contains messages for helping students when they make mistakes while learning or practicing.

Fig. 5. Learning modes samples: medium tension live line maintenance (left) and substations primary equiment testing (right)

These systems can be used for self learning or presential courses. In the former, all progress made by students is recorded locally in a database, so that students can learn dangerous procedures at their own pace with no risk at all. In the later, all progress is recorded in a server and this information is available for trainers and other company authorities. In this way, learning progress within

the company can be monitored. The evaluation mode includes two kinds of tests namely, theoretical tests, which are based on multiple choice questions, and practical tests in which a student have to perform a maintenance procedure, based on the same learning/practice 3D scenarios with no help from the system.

Currently, CFE is using successfully these systems for personnel training in all of its 16 distribution divisions located in the whole country. ALEn3D for medium line maintenance training was the winner of the CFE's INNOVA 2008 award. These systems have been developed within the first stage of the VR Roadmap [16]. In this stage, the development of non immersive VR systems [17] is planned. Currently, the GRV is starting the second stage in which the development of immersive systems, the inclusion of augmented reality and collaborative virtual environments are considered.

4.2 Visor3D Coupled to a Dynamic Simulator

In this section a prototype of a non-immersive 3D display system coupled to a dynamic real–time full–scope simulator is presented. The name of the prototype is Visor3D, whose aim is to create a virtual world representation of a section of a thermal power plant, with the equipment modeled and textured in detail, so that the user has a realistic sensation of being in the scene. Visor3D shows equipment in 3D and displays, in real time, the recorded status according to the initial condition set by the instructor (from the simulation station), while that one with local control can be infered directly from the simulation session. The main components of the prototype are the simulation system and the display system.

The Simulation System is composed of a set of integrated software components that were adapted to establish communication with Visor3D. A more detailed explanation about the simulator software is available in [18]. In the Display System, the Visor3D is responsible of rendering the virtual environment of the power plant, which consists of static and dynamic equipment (see Fig. 6). The former displays its status in the simulator in real time and can also perform actions such as opening / closing valves through internal parameters. These commands are sent in real time and have influence in the simulation session. The Visor3D also includes an auxiliary operator in the stage, which is represented by an avatar, and supports three types of navigation on the scene: in first person, third person and free form.

Additionally, another component was developed to communicate the simulator and the Visor3D prototype. Its objective is to establish communication with the display system and respond to requests of variables and commands that the user performs from the equipment shown in the virtual environment.

The development of this kind of systems, that include Virtual Reality and dynamic simulation, will help to reinforce the operator training sessions, since he/she will be able to observe in 3D, the effects of his/her operations and phenomena that occur in the processes of power generation that cannot be observed.

Fig. 6. Display System. Visor3D is responsible of rendering the virtual environment of the power plant

5 Training Simulators

5.1 Power Plant Simulators and Expert System for Operator Training

A power plant simulator is a software system that models the behavior of a real power plant and that can be used for the operators training of such systems. Since it allows the training in special situations that could arise in real life but without endanger persons or real equipment, its principal aim is to provide an integrated training avoiding unnecessary risks .

It has been verified that the operators training of power plants based on training simulators is the most effective way in which an operator familiarizes with the plant at which it will be employed. However, due the high cost of such a training, it is desirable that operators attend these sessions with some previous theoretical /practical knowledge.

The IIE has developed a simulator–expert system [19] aimed to provide a standalone application where operative personnel can practice and be evaluated without the supervision of an instructor. The practice will complement the regular training courses carried out at the training center. This system was developed for the CFE using a 350 MW Coal-Fired power plant simulator (CFS) and a 450MW Combined Cycle power plant simulator (CCS) .

The system is divided into two parts: the definition of an exercise and its execution (see Fig. 7). The exercise is defined by the instructor who stablishes the initial condition of the simulation system, creates questions for the theoretical evaluation, assigns the corresponding multimedia material for the theoretical lessons, and selects the process variables for the simulation evaluation. After that, a tool transforms the exercise into a set of rules to guide/track the trainee operation to be used by an expert system. During the exercise execution, the trainee has a graphical interface to perform a practice. This interface contains the

main operator's diagrams related to the exercise. The operator may be assisted by the system to indicate what actions to perform to complete the exercise correctly or just to monitor and report a wrong action.

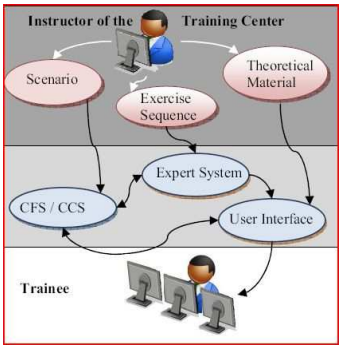

Fig. 7. Simulator–expert system. The system is composed of the definition of an exercise and its execution

6 Virtual Learning Environments

6.1 Virtual Postgraduate Center e-Learning Platform

The IIE Postgraduate Center (IIE-CP) has developed an e-learning platform called Virtual Postgraduate Center (CPV) [20]. It was implemented using a Learning Management System (LMS) which includes a number of activities and resources for on-line courses. CPV complies with the SCORM standard from Advanced Distributed Learning (ADL), so that SCORM 2004 4th Edition 1.1 learning objects (LO) were developed, which correspond to SCORM compliant teaching material Shareable Content Objects (SCO). In other words, the Content Aggregation Model (CAM - activity tree), the Run-Time Environment Model (RTE– Application Program Interface API between LMS and SCO), the Sequencing Definition Model (S) and the Navegation Model (N) were included into the SCOs using the Tracking Data Model (cmi. - Computer Managed Instruction) and the Sequencing Elements (adl.). The on-line courses structure is in accordance to SCORM and ADL recommendations. Therefore, SCORM content packages are automated by configuring the learning strategy with evaluations and remediation. For tracking the students performance individually and according to his/her perfomance, teaching materials are presented.

Content packages were developed using Author Tools for the e-learning teaching material. An instructional design team and a development team were integrated, and a script was developed to capture the information from instructors about how they want to show their teaching material.

An LMS offers asynchronous activities, such as discusion forums, and synchronous activities, such as a chat. It also manages the courses, ad hoc configurations, security, accounts and profiles, massive registration and by e-mail, and customizable views and templates. In each course, it registers every interaction and grades, and it generates reports and statistics. At present, CPV has several on–line courses on various topics related to the electric power sector.

As a future work, several new functionalities will be implemented. For instance, the HTML editor (with equations editor and calculator) will be included into the chat (for loading images, videos, audio-narration-music), an on-line author tool (LAMS Learning Activity Management System) for developing teaching material, a virtual classroom (videoconference, whiteboard, chat and desk sharing) and a remote laboratory (VNC-Virtual Network Connection and WebCam) will be integrated, SMS messages (Short Messaging Service) for the students' mobile phones from the LMS will be included, an intelligent tutor system (to integrate an intelligent LMS) will be implemented within the LMS, and a mobile learning tool will be added. In order to develop intelligent learning objects an intelligent sequencing will be implemented for the SCORM content packages.

7 Conclusions

A set of eight training and tutoring learning technologies developed at the Electrical Research Institute were presented. It included developments focused on providing different learning styles, considering the student affective state and enabling multi-functionality. Samples of assistant systems, virtual reality systems, power plant real simulators, and virtual center e-learning platforms were also presented.

The developments described here are a good example of how modern technologies provide a great variety of approaches to support the electric power industry.

References

1. Felder, R., Silverman, L.: Learning and teaching styles in engineering education. Engineering Education **78**(7) (1988) 674–681
2. Hernández, Y., Rodríguez, G.: Learning styles theory for intelligent learning environments: Adapting the instruction. In: 3rd International Conference on Computer Supported Education. Volume 1., SciTePress (2011) 456–459
3. Felder, R., A.Soloman, B.: Learning styles and strategies. Technical report, NCSU North Carolina State University (1993)
4. Savic, G., Konjovic, Z.: Learning style based personalization of scorm e-learning courses. In: 7th International Symposium on Intelligent Systems and Informatics, SISY 2009, IEEE (2009) 349–353
5. Ortony, A., Clore, G., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press (1988)
6. Hernández, Y., Sucar, L.E., Arroyo-Figueroa, G.: Evaluating an affective student model for intelligent learning environments. Iberamia 2010 (2010) 473–482

7. Costa, P., McCrae, R.: Four ways five factors are basic. Personality and Individual Differences **13**(1) (1992)
8. Hernández, Y., Sucar, L.E., Conati, C.: Incorporating an affective behavior model to an intelligent tutor. In Guesgen, H.W., Lane, H.C., eds.: FLAIRS Conference 2009, FLAIRS (2009) 448–453
9. Reyes, A., Ibarguengoytia, P., Elizalde, F., Sánchez, L., Nava, A.: ASISTO: An intelligent assistant system for power plant operation and training. In: 16th Intern. Conf. on Intelligent Systems Application to Power Systems,ISAP 2011, Crete Island, Greece (September 2011)
10. Puterman, M.: Markov Decision Processes. Wiley, New York (1994)
11. Elizalde, F., Sucar, L.E., Luque, M., Diez, J., Reyes, A.: Policy explanation in factored Markov decision processes. In: Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM 2008), Hirtshals, Denmark (September 2008) 97–104
12. Buen, P.D., Morales, E., Vadera, S.: A knowledge-based framework for learning, applying and consulting procedures. In C. Frasson, G.G.y.A.L., ed.: Intelligent Tutoring Systems, Germany, Third International Conference, ITS'96, Springer-Verlag (1996) pp. 392–400
13. de Buen, P., Morales, E., Vadera, S.: A collaborative approach to user modeling within a multi-functional architecture. In for Mechanical Sciences, I.C., ed.: CISM Courses and Lectures. Volume No. 407., Italy, Springer-Verlag Wien NewYork (1999) pp. 291–293
14. de Buen, P., Vadera, S., Morales, R.: Machine learning in LacePro multi-functional framework. In: Proceedings of the UM97 Workshop on Machine Learning for User Modeling, Chia Laguna, Sardinia, Italy, Sixth International Conference on User Modeling (June 1997)
15. Pérez, E.I., Rada, J.B., Lima, J.R., Marín, M.M.: Design and costs estimation of electrical substations based on three-dimensional building blocks. 6th International Symposium on Visual Computing (2010)
16. Pérez, M.: Ruta tecnológica de realidad virtual para el sector eléctrico. In Meléndez, A.M., ed.: Komputer Sapiens. Revista de Divulgación de la SMIA.2011. Volume 1., Reforma 113. Col. Palmira. Cuernavaca Mor., México (2011) 11–16
17. Pérez, M., Zabre, E., Islas, E.: Realidad virtual: Un panorama general. In: Boletín IIE:2004, Reforma 113. Col. Palmira. Cuernavaca Mor., México (2004) 39–44
18. Tavira, J., Jimenez, L., Romero, G.: A simulator for training fossil-fuel power plants operators with an hmi based on a multi-window system. International Journal of Computer Aided Engineering and Technology 2010 **2**(1) (2010) 30–40
19. Tavira, J., Martnez, R., Jiménez, F.: Power plants simulators with an expert system to train and evaluate operators. In: Proceedings of the World Congress on Engineering and Computer Science 2010. Volume Vol II., San Francisco, USA, WCECS 2010 (October 20-22 2010)
20. Jiménez-Fraustro, F.F., García-Hernández, C.F., Aguilar-Figueroa, S.A., Martínez-Ramírez, R.E.: Development of the IIE's virtual postgradute center: Phase i. Technical report, IIE (December 2010)

Regular Papers

Meaning Representation for Automatic Extraction of Lexical Functions

Olga Kolesnikova

Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Mexico, D.F. 07738, Mexico.
kolesolga@gmail.com

Resumen. Lexical functions formalize semantic and syntactic relations between lexical units, given that meaning of an individual word largely depends on various relations connecting it to other words in context. Collocational relation is a type of institutionalized lexical relations that holds between the base and its partner in a collocation in contrast to free word combination where both words are used in their typical meaning. Collocation are important for natural language processing because collocation comprises the restrictions on how words can be used together. The formalism of lexical functions is a means of representing such information. If collocations are annotated with lexical functions in a computer readable dictionary, it allows effective use of collocations in natural language applications including parsers, high quality machine translation, periphrasis systems and computer-aided learning of lexica. In order to create such applications, we need to extract lexical functions from corpora automatically. For this, we represent the lexical meaning of a given word with a set of all its hypernyms extracted from the Spanish WordNet.

Keywords: natural language processing, lexical functions, semantic representation, machine learning.

1 Introducción

Lexical function is a concept that formalizes semantic and syntactic relations between lexical units. Relations between words are a vital part of any natural language system. Meaning of an individual word largely depends on various relations connecting it to other words in context. In particular, collocational relation is a type of institutionalized lexical relations that holds between the base and its partner in a collocation.

A collocation typically consists on the main word, or base, and the word that collocates with it, or collocate. Examples of collocations are *give a lecture*, *make a decision*, *lend support*, where the bases are *lecture*, *decision*, *support* and the partners, termed collocates, are *give*, *make*, *lend*. Collocations are opposed to free word combination where both words are used in their typical meaning, such as, for example, *give a book*, *make a dress*, *lend money*.

Knowledge of collocation is important for natural language processing and its applications because collocation comprises the restrictions on how words can be used together. There are many methods to extract collocations automatically but their result is a plain list of collocations. Such lists are more valuable if collocations are tagged with semantic and grammatical information.

The formalism of lexical functions is a means of representing such information. If collocations are annotated with lexical functions in a computer readable dictionary, it will allow effective use of collocations in natural language applications including parsers [Gelbukh and Sidorov 2006, Bolshakov and Gelbukh 2004], high quality machine translation [Bolshakov and Gelbukh 2001], periphrasis system and computer-aided learning of lexica [Bolshakov and Gelbukh 2002].

In order to create such applications, we need to extract lexical functions from corpora automatically. It is our intent to extract Spanish verb-noun collocations belonging to a given lexical function from corpora. To achieve this, we represent the lexical meaning of a given word with a set of all its hypernyms. This allows us to use machine-learning techniques for predicting lexical functions as values of the class variable for unseen collocations. We extract such hypernyms from the Spanish WordNet. Our experiments show that machine learning is feasible to achieve the task of automatic detection of lexical functions.

Relatively little research has been done so far on automatic detection of lexical functions. In fact, there are only two papers that report results on performance of a few machine-learning algorithms on classifying collocations according to the typology of lexical functions [Wanner 2004, Wanner *et al*. 2006].

In this paper, we first review various definitions of collocations given in existing literature, to clarify thoroughly the goal of our research. Then we summarize existing work on collocation extraction, in particular on extraction of lexical functions: we consider the work done in [Wanner 2004], [Wanner *et al*. 2006] and comment on another research on automatic extraction of lexical functions [Alonso Ramos *et al*. 2008] based on an approach different from the work in [Wanner 2004] and [Wanner *et al*. 2006]. We discuss the three statements, or hypotheses, made in [Wanner *et al*. 2006]. Next, we describe the data used in our experiments. Finally, we present our methodology for meaning representation for collocation extraction.

1.1 Definitions of Collocation

Our goal is to extract lexical functions, which are a particular type of a collocation. There is no consensus on what a collocation is. Here we review numerous different approaches to the definition of a collocation.

With each definition, additional information is given as to the source of definition, the criterion used to distinguish collocations from free word combinations, and some our comments on the particular definition.

[Firth 1957]: *Collocations of a given word are statements of the habitual or customary places of that word*. Lexical criterion: a word is used in a fixed position with respect to a given word; statistical criterion: frequency of word co-occurrence.

[Firth 1957] first introduced the term 'collocation' from Latin *collocatio* which means 'bringing together, grouping'. Firth believes that speakers make 'typical' common lexical choices in collocational combinations. Collocation is a concept in Firth's theory of meaning: "Meaning by collocation is an abstraction at the syntagmatic level and is not directly concerned with the conceptual or idea approach to the meaning of words. One of the meanings of *night* is its collocability with *dark*, and of *dark*, of course, collocation with *night*."

[Halliday 1961]: *Collocation is the syntagmatic association of lexical items, quantifiable, textually, as the probability that there will occur, at n removes (a distance of n lexical items) from an item x, the items a, b, c ...* Lexical criterion: a word is used a fixed position with respect to a given word. Statistical criterion: high co-occurrence frequency.

If a lexical item is used in the text, then its collocate has the highest probability of occurrence at some distance from the lexical item. Collocations cut across grammar boundaries: e.g., *he argued strongly* and *the strength of his argument* are grammatical transformations of the initial collocation *strong argument*.

[Hausmann 1984]: *Collocations are binary word-combinations, consist of words with limited combinatorial capacity, they are semi-finished products of language, affine combinations of striking habitualness. In a collocation one partner determines, another is determined. In other words: collocations have a basis and a co-occurring collocate*. Lexical criterion: the lexical choice of the collocate depends on the basis.

Word combinations are classified word-combinations according to the features fixed vs. non-fixed, and in this classification collocations are belong to the category of non-fixed affine combinations. Internal structure of collocation: collocation components have functions of a basis and a collocate, and the basis (not the speaker) 'decides' what the collocate will be.

[Benson 1986]: *Collocation is a group of words that occurs repeatedly, i. e. recurs, in a language. Recurrent phrases can be divided into grammatical collocations and lexical collocations. Grammatical collocations consist of a dominant element and a preposition or a grammatical construction: fond of, (we reached) an agreement that... Lexical collocations do not have a dominant word, their components are "equal": to come to an agreement, affect deeply, weak tea.* Functional criterion: collocations are classified according to function of collocational elements. Statistical criterion: high co-occurrence frequency.

This is a broad understanding of collocation. Classification of collocations according to their compositional structure is given.

[Benson 1990]: *Collocations should be defined not just as 'recurrent word combinations', <but as> 'ARBITRARY recurrent word combinations'*. Lexical criterion: arbitrariness and recurrency.

'Arbitrary' here is opposed to 'regular' means that collocations are not predictable and cannot be translated word by word.

[Van Roey 1990]: *Collocation is "that linguistic phenomenon whereby a given vocabulary item prefers the company of another item rather than its 'synonyms' because of*

constraints which are not on the level of syntax or conceptual meaning but on that of usage." Statistical criterion: high co-occurrence frequency in corpora.

Van Roey summarizes statistical view stated by Halliday in terms of expression or 'usage'. A collocate can thus simply be seen as any word which co-occurs within an arbitrary determined distance or *span* of a central word or *node* at the frequency level at which the researcher can say that the co-occurrence is not accidental. This approach is also textual in that it relies solely on the ability of the computer program to analyze large amounts of computer-readable texts.

[Cowie 1994]: *Collocations are associations of two or more lexemes (or roots) recognized in and defined by their occurrence in a specific range of grammatical constructions*. Structural criterion: collocations are distinguished by patterns.

Collocations are classified by Cowie into types according to their grammatical patterns.

[Howarth 1996]: *In his lexical continuum model, collocations as composite units are placed on a sliding scale of meaning and form from relatively unrestricted (collocations) to highly fixed (idioms). Restrictive collocations are fully institutionalised phrases, memorized as wholes and used as conventional form-meaning pairings*. Syntactic criterion: commutability – the extent to which the elements in the expression can be replaced or moved (make/reach/take decision vs. shrug one's shoulders). Semantic criterion: motivation – the extent to which the semantic origin of the expression is identifiable (move the goalposts = to change conditions for success vs. shoot the breeze = to chatter, which is an opaque idiom).

Classification includes four types of expressions with no reference to frequency of occurrence:

free collocation: *blow a trumpet* = to play a trumpet,
restrictive collocation: *blow a fuse* = to destroy a fuse/to get angry,
figurative idiom*: blow your own trumpet* = to sell oneself excessively,
pure idiom: *blow the gaff* = to reveal a concealed truth.

The problem with this classification is that is difficult to determine what is meant by 'syntactically fixed', 'unmotivated' or 'opaque'. This is seen in the ambiguous example of *to blow a fuse*.

[Sinclair et al. 2004]: *Collocation is the co-occurrence of two items in a text within a specified environment. Significant collocation is regular collocation between two items, such that they co-occur more often than their respective frequencies. Casual collocations are "non-significant" collocations*. Lexical criterion: recurrency of co-occurrence. Statistical criterion: high co-occurrence frequency.

The degree of significance for an association between items is here determined by such statistic tests as Fischer's Exact Test or Poisson Test.

[Mel'čuk 1996]: *Collocation is a combination of two lexical items in which the semantics of one of the lexical items (the base) is autonomous from the combination it appears in, and where the other lexical item (the collocate) adds semantic features to the semantics of the base. [Gledhill 2000] explains that for Mel'čuk a collocation is a semantic function operating between two or more words in which one of the words keeps its*

'normal' meaning. Semantic criterion: the meaning of a collocation is not inferred from the meaning of the base combined with meaning of the collocate.

According to Mel'čuk, semantics of a collocation is not the meaning of the base + the meaning of the collocate, but rather the meaning of the base + some additional meaning that are included in the meaning of the base. In particular: '...the concept of collocation is independent of grammatical categories: the relationship, which holds between the verb *argue* and the adverb *strongly* is the same as that holding between the noun *argument* and the adjective *strong*' [Fontenelle 1994].

2 Related Work

Bolshakov and Gelbukh [1998] studied lexical functions in Spanish on a number of examples. They described various types of such collocations and discussed how some functions can be combined to give rise to new meanings. They also studied classification of collocations according to the meaning of the words being combined [Bolshakov and Gelbukh, 2000].

In 2004, L. Wanner proposed to view the task of LF detection as automatic classification of collocations according to LF typology. To fulfill this task, the nearest neighbor machine learning technique was used. Datasets included Spanish verb-noun pairs annotated with nine LFs: $CausFunc_0$, $Caus_2Func_1$, $IncepFunc_1$, $FinFunc_0$, $Oper_1$, $ContOper_1$, $Oper_2$, $Real_1$, $Real_2$. Verb-noun pairs were divided in two groups. In the first group, nouns belonged to the semantic field of emotions; in the second groups nouns were field-independent. As a source of information for building the training and test sets, hypernymy hierarchy of the Spanish part of EuroWordNet was used.

The words in the training set were represented by their hypernyms, Basic Concepts and Top Concepts. The average *F-measure* of about 70% was achieved in these experiments. The best result for field-independent nouns was F-measure of 76.58 for $CausFunc_0$ with the meaning 'cause the existence of the situation, state, etc.' The Causer is the subject of utterances with $CausFunc_0$.

In [Wanner *et al*. 2006], four machine learning methods were applied to classify Spanish verb-noun collocations according to LFs, namely Nearest Neighbor technique, Naïve Bayesian network, Tree-Augmented Network Classification technique and a decision tree classification technique based on the ID3-algorithm. As in [Wanner 2004], experiments were carried out for two groups of verb-noun collocations: nouns of the first group belonged to the semantic field of emotions; nouns of the second group were field-independent. Lexical functions were also identical with [Wanner 2004] as well as data representation. The best results for field-independent nouns were shown by ID3 algorithm (F-measure of 0.76) for $Caus_2Func_1$ with the meaning 'cause something to be experienced / carried out / performed', and by the Nearest Neighbor technique (F-measure of 0.74) for $Oper_1$ with the meaning 'perform / experience / carry out something'. The Causer is the subject of utterances with $Caus_2Func_1$, and the Agent is the direct object of the verb which is the value of $Cuas_2Func_1$. In utterances with $Oper_1$, the Agent is the subject.

As we are interested in experiments with verb-noun collocations where the nouns have various semantics, i.e., the nouns are field-independent, Tables 1–4 summarizes the results for field-independent nouns only in [Wanner 2004] and [Wanner *et al*. 2006].

Table 1 gives the meaning of lexical functions used in experiments only with field-independent nouns [Wanner 2004]. We give examples in Spanish with literal translation in English. After the name of a lexical function, we give three figures with the following meaning:

- the number of examples of a given LF in the training set;
- the number of examples of a given LF in the test set;
- the total number of examples of a given LF in the training set and in the test set.

Table 1. Data in [Wanner 2004]

Name	Meaning	Examples in Spanish	Lit. translation in English
$Oper_1$ 35 + 15 = 50	*experience, perform, carry out something*	*dar golpe* *presentar una demanda* *hacer campaña* *dictar la sentencia*	*give a blow* *present a demand* *make a campaign* *dictate a sentence*
$Oper_2$ 33 + 15 = 48	*undergo, be source of*	*someterse a un análisis* *afrontar un desafío* *hacer examen* *tener la culpa*	*submit oneself to analysis* *face a challenge* *make exam* *have guilt*
$CausFunc_0$ 38 + 15 = 53	*cause the existence of the situation, state, etc.*	*dar la alarma* *celebrar elecciones* *provocar una crisis* *publicar una revista*	*give the alarm* *celebrate elections* *provoke a crisis* *publish a magazine*
$Real_1$ 37 + 15 = 52	*act accordingly to the situation, use as forseen*	*ejercer la autoridad* *utilizar el teléfono* *hablar la lengua* *cumplir la promesa*	*exercise authority* *use a telephone* *speak a language* *keep a promise*
$Real_2$ 38 + 15 = 53	*react accordingly to the situation*	*responder a objeción* *satisfacer un requisito* *atender la solicitud* *rendirse a persuasión*	*respond to an objection* *satisfy a requirement* *attend an application* *surrender to persuasion*

Table 2 lists LFs with respective number of examples in [Wanner *et al*. 2006] for verb-noun combinations with field-independent nouns.

Table 2. Data in [Wanner *et al*. 2006]

LF	Number of Examples
$CausFunc_0$	53
$Oper_1$	87
$Oper_2$	48
$Real_1$	52
$Real_2$	53

Table 3 presents the results reported in the referenced paper, in terms of accuracy by each lexical function.

Table 3. Results in [Wanner 2004]

F-measure/LF	**$CausFunc_0$**	**$Oper_1$**	**$Oper_2$**	**$Real_1$**	**$Real_2$**
field-independent nouns	76.58	60.93	75.85	74.06	58.32

Finally, Table 4 shows the results in [Wanner *et al*. 2006]; the values of precision, recall and F-measure are given in the following format: <precision> | <recall> | <F-measure>. Not all four machine-learning methods in Table 4 were applied to all LFs; if experiments were not made for a particular method and LF, N/A is put instead of precision, recall, and F-measure.

Table 4. Results in [Wanner *et al*. 2006]

LF	**Machine learning technique**			
	NN	**NB**	**TAN**	**ID3**
$CausFunc_0$	0.59 | 0.79 | 0.68	0.44 | 0.89 | 0.59	0.45 | 0.57 | 0.50	N/A
$Caus_2Func_1$	N/A	N/A	N/A	0.53 | 0.65 | 0.50
$FinFunc_0$	N/A	N/A	N/A	0.53 | 0.40 | 0.40
$IncepFunc_1$	N/A	N/A	N/A	0.40 | 0.48 | 0.40
$Oper_1$	0.65 | 0.55 | 0.60	0.87 | 0.64 | 0.74	0.75 | 0.49 | 0.59	0.52 | 0.51 | 0.50
$Oper_2$	0.62 | 0.71 | 0.66	0.55 | 0.21 | 0.30	0.55 | 0.56 | 0.55	N/A
$ContOper_1$	N/A	N/A	N/A	0.84 | 0.57 | 0.68
$Real_1$	0.58 | 0.44 | 0.50	0.58 | 0.37 | 0.45	0.78 | 0.36 | 0.49	N/A
$Real_2$	0.56 | 0.55 | 0.55	0.73 | 0.35 | 0.47	0.34 | 0.67 | 0.45	N/A

On the other hand, [Alonso Ramos et al. 2008] proposed an algorithm for extracting collocations following the pattern "support verb + object" from FrameNet corpus of examples [Ruppenhofer *et al*. 2006] and checking if they are of the type $Oper_n$. This work takes advantage of syntactic, semantic and collocation annotations in the FrameNet corpus, since some annotations can serve as indicators of a particular LF. The authors tested the proposed algorithm on a set of 208 instances. The algorithm showed accuracy of 76%. The researchers conclude that extraction and semantic classification of collocations is feasible with semantically annotated corpora. This statement sounds logical because the formalism of lexical function captures the correspondence between the semantic valence of the keyword and the syntactic structure of utterances where the keyword is used in a collocation together with the value of the respective LF.

2.1 Hypothesis Stated by Wanner et al.

Wanner *et al*. [2006] experimented with the same type of lexical data as in [Wanner 2004], i.e. verb-noun pairs. The task is to answer the question: what kind of collocational features are fundamental for human distinguishing among collocational types. The authors view collocational types as LFs, i.e. a particular LF represents a certain type of collocations.

Three hypotheses are put forward as possible solutions, and to model every solution, an appropriate machine learning technique is selected. Below we list the three hypotheses and the selected machine learning techniques.

1. Collocations can be recognized by their similarity to the prototypical sample of each collocational type; this strategy is modeled by the Nearest Neighbor technique.
2. Collocations can be recognized by similarity of semantic features of their elements (i.e., base and collocate) to semantic features of elements of the collocations known to belong to a specific LF; this method is modeled by Naïve Bayesian network and a decision tree classification technique based on the ID3-algorithm.
3. Collocations can be recognized by correlation between semantic features of collocational elements; this approach is modeled by Tree-Augmented Network Classification technique.

It should be mentioned, that having proposed three hypotheses, the authors have not yet demonstrated their validity by comparing the performance of many machine-learning techniques known today, but applied only four learning algorithms to illustrate that three human strategies mentioned above are practical.

2.2 Automatic Detection of Semantic Relations

There has been some research done on semantic relations in word combinations, for example, one that deals with automatic assignment of semantic relations to English noun-modifier pairs in [Nastase and Szpakowicz 2003, Nastase *et al*. 2006]. Though in our work, verb-noun combinations are treated, we believe that the principles of choosing data representation and machine learning techniques for detection of semantic relations between a noun and a modifier can be are used to detect semantic relations in verb-noun pairs.

The underlying idea is the same: learning the meaning of word combinations. In [Nastase and Szpakowicz 2003, Nastase *et al*. 2006], the researchers examined the following relations: causal, temporal, spatial, conjunctive, participant, and quality. They used two different data representations: the first is based on WordNet relations, the second, on contextual information extracted from corpora. They applied memory-based learning, decision tree induction and Support Vector Machine. The highest F-measure of 0.847 was achieved by C5.0 decision tree to detect temporal relation based on WordNet representation.

3 Data Sets Used in Our Experiments

We have created a unique lexical resource of Spanish lexical functions in order to compile training sets for machine learning experiments.

3.1 Data for the Training Sets

For training and for testing, we used in our experiments the data sets presented in the sequel.

Lexical Resources

Lexical resources are widely used in natural language processing and their role is difficult to overestimate. Lexical resources vary significantly in language coverage and linguistic information they include, and have many forms: word lists, dictionaries, thesauri, ontologies, glossaries, concordances, etc.

For Spanish, this diversity of forms can be illustrated with the following lexicographic works:

- A Medieval Spanish Word List [Oelschläger 1940],
- Diccionario de la Lengua Española (Dictionary of the Spanish Language) [RAE 2001],
- Streetwise Spanish Dictionary/Thesaurus [McVey and Wegmann 2001],
- Spanish part of EuroWordNet [Vossen 1998], an electronic lexical onthology,
- Glosario de voces comentadas en ediciones de textos clásicos [Fontecha 1941],
- Concordancia electrónica de la Biblia online (for Reina Valera version, 1960) [CEB].

Machine-readable resources are of special interest, since they comprise an integral part of computer systems aimed at automatic language treatment and language generation.

Though computerized lexicography has achieved a significant progress over last years, compilation of high quality dictionaries still requires a lot of manual work. In such a multi-faceted area as computational linguistics, it is difficult sometimes to find an adequate lexical resource (and for the language you need) for a specific research task or application.

One way to solve this problem is to develop computational procedures that can adjust existing resources to the demands of a researcher. However, this is not always effective. Certainly, the best solution of this problem is to compile a new lexical resource, but this is not always feasible in view of its cost.

We present a list of most frequent Spanish verb-noun pairs which contains semantically annotated collocations and free word combinations. It is a machine readable lexical resource where each verb-noun pair is associated with the following linguistic data:

1. whether a pair is a free word combination or a collocation;
2. if a verb-noun pair is a collocation, it is marked with lexical functions;
3. word senses of the Spanish WordNet [Vossen 1998, SpWN] are assigned to both elements of the verb-noun pair.

Existing Lexical Resources

A number of lexical resources contain lexical functions. Almost all of them are not specialized dictionaries of lexical functions, but include lexical functions together with other linguistic information.

The concept of lexical function was originally proposed by researchers of the Russian semantic school. Lexical functions have been applied there for description of lexica and machine translation. A dictionary in Russian compiled by Apresjan [referenced in

Apresjan 2004] for the machine translation system ETAP includes more than 100 lexical functions with definitions and examples. For instance, for the verbal lexical function $Oper_1$, the dictionary contains several hundreds of samples.

Lexical functions are used to describe the word's combinatory power in Explanatory Combinatorial Dictionaries compiled for Russian [Mel'čuk and Zholkovskij 1984] and for French [Mel'čuk *et al*. 1984, 1988]. For every word, its lexical entry includes a list of lexical functions applicable to it with their respective values. For French, an on-line dictionary, the DiCo, is referenced in [Wanner 2004] but we could not access it on the web.

For Spanish, there exists a dictionary of collocations, Diccionario de colocaciones del Español [DiCE] [Alonso Ramos 2003] annotated with lexical functions, but the DiCE is limited only to nouns belonging to the semantic field of emotions. [Sanromán 1998, 2003] compiled collections of Spanish collocations also for emotion nouns classified in terms of lexical functions. [Wanner 2004, Wanner *et al*. 2006] used Sanromán's collections for machine learning experiments, and for the same purpose, compiled additional lists of Spanish verb-noun collocations annotated with lexical functions. In the additional lists nouns were semantically field independent. The overall number of LF instances in the latter lists were 256 [Wanner 2004] and 293 [Wanner *et al*. 2006]. Unfortunately, these lists are no longer available in full.

Description of the Lexical Resource

In this section, we describe the lexical resources that we used for our experiments on automatic extraction of lexical functions from raw text corpora.

Compilation Firstly, the Spanish Web Corpus [SpWC] was chosen as a source of verb-noun pairs with the pattern verb + direct object. All such verb-noun pairs used in the Spanish Web Corpus five or more times, were extracted automatically from the said corpus by the Sketch Engine [Kilgarriff *et al*. 2004], a web-based program for corpus processing. Fig. 1 displays the interface of the Sketch Engine where several corpora are listed including the Spanish Web Corpus. The obtained list contained 83,982 verb-noun pairs, and it was ranked by frequency.

Secondly, one thousand pairs were taken from the upper part of the list, i.e. most frequent verb-noun pairs.

Thirdly, in the list of one thousand pairs, erroneous combinations were marked with the label ERROR. Erroneous pairs included, for instance, past participle or infinitive instead of noun, or contained symbols like --, « , © instead of words. How did errors emerge? The automatic extraction procedure was set to search for combinations with the pattern verb + direct object in the corpus. This procedure needs part of speech (POS) and lemma information, and such data is supplied by TreeTagger, software used to annotate the Spanish Web Corpus with POS and lemmas. The TreeTagger is a leading tool applied for POS tagging and lemmatization, it achieves high accuracy but still is error-prone. Due to errors made by the TreeTagger, the set of extracted verb-noun pairs contained fallacious combinations. For the sake of preserving the original design of automatically extracted set, these incorrect combinations were not removed from the list but identified as wrong. The total number of erroneous pairs was 61, so after their removal the list contained 939 pairs.

Fourthly, collocational verb-noun pairs were annotated with lexical functions. The rest of the pairs were annotated as free word combinations using the label FWC.

Lastly, all verbs and nouns in the list were disambiguated with word senses from the Spanish WordNet, an electronic lexicon structured the same way as WordNet for English. For some verb-noun pairs, relevant senses were not found in the above-mentioned dictionary, and the number of such pairs was 39. For example, in the combinaiton *dar cuenta*, *give account*, the noun *cuenta* means razón, *satisfacción de algo* (*reason*, *satisfaction of something*). This sense of *cuenta* is taken from Diccionario de la Lengua Española (Dictionary of the Spanish Language) [RAE 2001]. Unfortunately, this sense is absent in the Spanish WordNet so the expression *dar cuenta* was left without sense annotation. All such words were annotation N/A, i.e. not available.

Fig 1. Sketch Engine with the Spanish Web Corpus.

The annotated list was formatted as a table and saved in an MS Excel file. Fig. 2 shows the process of the compilation of the lexical resource schematically.

Contents of the lexical resource A partial representation of the list is given in Table 5; Table 6 lists all lexical functions found in the list of 1000 most frequent verb-noun

pairs, their frequencies in the Spanish Web Corpus, and the number of examples for each of them.

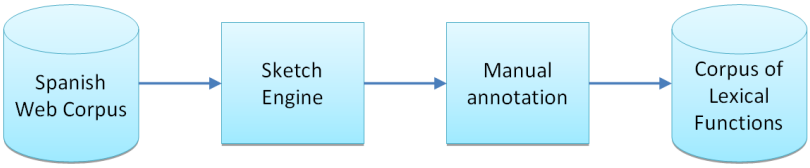

Fig. 2. The process of lexical resource compilation.

3.2 Data for the Test Sets

To build the test set, we extracted all verb-noun pairs from a corpus other than the corpus used to construct the training sets. So the data for test sets was mined from the Spanish Treebank Cast3LB [Civit and Martí 2004]. The number of all verb-noun pairs extracted from Cast3LB was 5181. We constructed four test sets, including, respectively, 100%, 75%, 50%, and 25% of all verb-noun pairs taken from Treebank Cast3LB.

Table 5. Partial representation of the lexical resource

LF/ FWC/ ERROR	VERB	Verb Sense Number	NOUN	Noun Sense Number	FREQ
$Oper_1$	dar	2	cuenta	N/A	9236
$CausFunc_0$	formar	2	parte	1	7454
$Oper_1$	tener	1	lugar	4	6680
$Oper_1$	tener	1	derecho	1	5255
$CausFunc_1$	hacer	2	falta	N/A	4827
$CausFunc_1$	dar	9	lugar	4	4180
$Oper_1$	hacer	15	referencia	2	3252
$Func_0$	hacer	N/A	año	2	3211
$Oper_1$	tener	1	problema	7	3075
$Func_0$	hacer	N/A	tiempo	1	3059
$IncepOper_1$	tomar	4	decisión	2	2781
$Oper_1$	tener	1	acceso	3	2773
$Oper_1$	tener	1	razón	2	2768
$Caus_2Func_1$	llamar	8	atención	1	2698
$Oper_1$	tener	1	sentido	1	2563
ERROR	haber	ERROR	estado	ERROR	2430
FWC	hacer	6	cosa	3	2374
$Oper_1$	tener	3	miedo	1	2226
ERROR	haber	ERROR	hecho	ERROR	2168

We did not disambiguate verb-noun pairs for the test sets manually. Instead, for each verb-noun, we built all possible verb-noun combinations of all senses in the Spanish WordNet. As an example, let us consider the pair *representar papel*, lit. *represent role*. The verb *representar* has 12 senses in the Spanish WordNet, and the noun *papel*, 5. This gives totally 60 combinations of *representar* and *papel* (12 multiplied by 5). The initial list for the test set included 5,181 verb-noun pairs, which resulted in totally 96,079 instances in the test set. A partial representation of the list is given in Fig. 3.

Table 6. Lexical functions with their respective frequency in corpus and the number of instances in the list of verb-noun pairs

LF	Freq	#	LF	Freq	#
$Oper_1$	165319	280	$PerfFunc_0$	1293	1
FWC	70211	202	$Caus_1Oper_1$	1280	2
$CausFunc_1$	45688	90	$Caus_1Func_1$	1085	3
$CausFunc_0$	40717	112	$IncepFunc_0$	1052	3
ERROR	26316	61	$PermOper_1$	910	3
$Real_1$	19191	61	$CausManifFunc_0$	788	2
$Func_0$	17393	25	$CausMinusFunc_0$	746	3
$IncepOper_1$	11805	25	$Oper_3$	520	1
$Oper_2$	8967	30	$LiquFunc_0$	514	2
$Caus_2Func_1$	8242	16	$IncepReal_1$	437	2
$ContOper_1$	5354	16	$Real_3$	381	1
Manif	3339	13	$PlusOper_1$	370	1
Copul	2345	9	$CausPerfFunc_0$	290	1
$CausPlusFunc_0$	2203	7	$AntiReal_3$	284	1
$Func_1$	1848	4	$MinusReal_1$	265	1
$PerfOper_1$	1736	4	$AntiPermOper_1$	258	1
$CausPlusFunc_1$	1548	5	$ManifFunc_0$	240	1
$Real_2$	1547	3	$CausMinusFunc_1$	229	1
$FinOper_1$	1476	6	$FinFunc_0$	178	1

```
v tener 1            n aire 11            n error 3
n aire 1             v tener 9            v salir 1
v tener 1            n aire 12            n error 4
n aire 2             v salir 1            v salir 1
...                  n error 1            n error 5
v tener 9            v salir 1            v salir 1
n aire 10            n error 2            n error 6
v tener 9            v salir 1            ...
```

Fig. 3. A part of the list of verb-noun pairs used for building the test set.

4 A New Method of Meaning Representation

In this section, we present our novel method for representing the meaning of words suitable for mining for lexical functions.

4.1 Data Representation

Each verb-noun pair in the training set and in the test set is represented as a set of all hypernyms of the noun and all hypernyms of the verb. The noun and the verb of the verb-noun pair were considered as zero-level hypernyms and thus were included in the set of hypernyms.

Hypernyms and Hyponyms

In linguistics, a hyponym is a word or phrase whose meaning is included within the meaning of another word, its hypernym. To put it simpler, a hyponym shares a type-of relationship with its hypernym. For example, *restaurant*, *rest house*, *planetarium*, *observatory*, *packinghouse*, *outbuilding*, *Pentagon* are all hyponyms of *building* (their hypernym), which is, in turn, a hyponym of *construction*.

In computer science, the relationship of hypernymy is often termed an "is-a" relationship. For example, the phrase *Restaurant is a building* can be used to describe the hyponymic relationship between *restaurant* and *building*.

Thus, hypernymy is the semantic relation in which one word is the hypernym of another one.

Spanish WordNet as a Source of Hypernyms

The Spanish WordNet follows the EuroWordNet [Vossen 1998] framework and is structured in the same way as the American WordNet for English [Miller 1998], namely, in terms of synsets (sets of synonymous words) with basic semantic relations between them.

Spanish nouns and verbs are organized into synonym sets, each representing one underlying lexical concept. Different relations, for example, hypernym relations, link the synonym sets.

Since all verbs and nouns have been disambiguated, hypernyms can be found for each word that has been annotated with its sense of the Spanish WordNet [SpWN]. Hypernyms were extracted automatically from the database of the dictionary referenced above. Fig. 4 and Fig. 5 display the interface of the Spanish WordNet as it is seen on the web. In the interface, we see hypernyms of *gato* “cat”.

Hypernyms as a Meaning Representation

A difference between data representation in our experiments and data sets used in [Wanner *et al*. 2006] should be noted here. In the paper just referenced, every word in the training set was accompanied by its synonyms and hypernyms, its own Base Concepts (BC) and the BCs of its hypernyms, its own Top Concepts (TC) and the TCs of its hypernyms taken from the Spanish part of the EuroWordNet [Vossen 1998]. Unlike those experiments, in our work for the sake of simplicity we included only hypernyms in our training sets.

Though in this case the data is annotated with less information, i.e. only with hypernyms, or in other words, only hypernyms are used to represent the meaning of verb-noun pairs, we hope that hypernyms would be sufficient to distinguish between lexical functions. Up to now, there has not been any research done that compares different data

representations for the task of predicting lexical functions of verb-noun pairs. Here we can remember the original intent of WordNet compilers [Miller 1998] who claimed that the meaning of any word could be described sufficiently well for at least human understanding by semantic relations only, like "is-a-kind-of" semantic relation of hypernym hierarchy.

Later, the authors of WordNet admitted that their previous assumption had been wrong and glosses were added to distinguish synonym sets. Though practical significance of glosses is generally accepted, we intent to study how well the meaning of lexical functions can be distinguished if only hypernym information is taken into account.

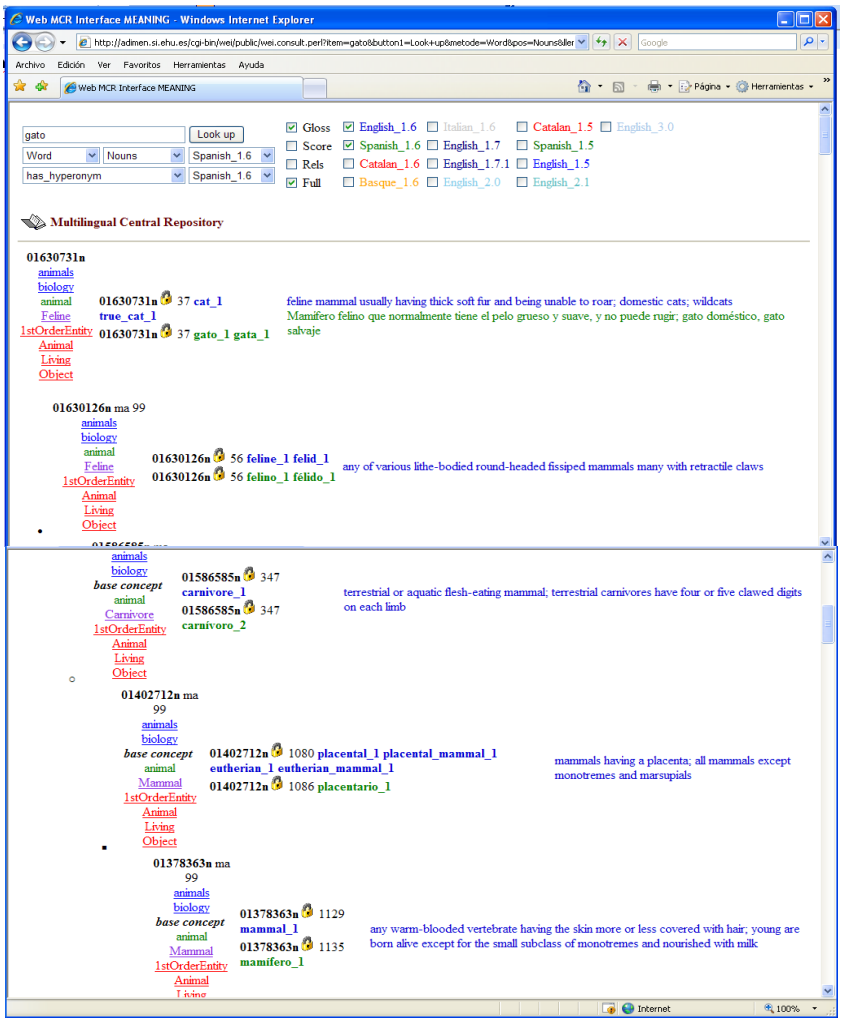

Fig. 4. The Spanish WordNet, hyperonyms for *gato*, cat.

Further research is needed to investigate how information other that hypernym taxonomy, for example, that of semantic ontologies, changes the performance of machine learning algorithms.

4.2 Linguistic Description of Training Sets and Test Sets

Lexical Functions Chosen for Experiments

Our choice of lexical functions depends on the number of examples that each lexical function has in the lexical resource of Spanish lexical functions created by us and described in Section 4.1. We have selected LFs that have the number of examples sufficient for machine learning experiments. [Wanner 2004] and [Wanner *et al*.2006] experimented with the following number of LF examples: the biggest number of examples that this researcher had in the training set was 87 for $Oper_1$ and the least number of examples was 33 for $Oper_2$.

Table 7. Lexical functions chosen for the experiments

LF and # of examples	Meaning	Collocation: LF value + keyword	
		Spanish	**English translation**
$Oper_1$ 280	Lat. *operare* – 'to do, perform'. Experience (if K is an emotion), carry out K.	*alcanzar un objetivo* *aplicar una medida* *corregir un error* *satisfacer una necesidad*	*achieve a goal* *apply a measure* *correct a mistake* *satisfy a necessity*
$CausFunc_0$ 112	Lat. *causare* – 'to cause'. Do something so that K begins occurring.	*encontrar respuesta* *establecer un sistema* *hacer campaña* *producir un efecto*	*find an answer* *establish a system* *conduct a campaign* *produce an effect*
$CausFunc_1$ 90	A person/object, different from the agent of K, does something so that K occurs and has effect on the agent of K.	*abrir camino* *causar daño* *dar respuesta* *producir un cambio*	*open the way* *cause damage* *give an answer* *produce a change*
$Real_1$ 61	Lat. *realis* – '*real*'. To fulfill the requirement of K, to act according to K.	*contestar una pregunta* *cumplir el requisito* *solucionar un problema* *utilizar la tecnología*	*answer a question* *fulfill the requirement* *solve a problem* *use technology*
$Func_0$ 25	Lat. *functionare* – 'to function'. K exists, takes place, occurs.	*el tiempo pasa* *hace un mes* *una posibilidad cabe* *la razón existe*	*time flies* *a month ago* *there is a possibility* *the reason exists*
$Oper_2$ 30	Undergo K, be source of K	*aprender una lección* *obtener una respuesta* *recibir ayuda* *sufrir un cambio*	*learn a lesson* *get an answer* *receive help* *suffer a change*

$IncepOper_1$ 25	Lat. *incipere* – 'to begin'. Begin to do, perform, experience, carry out K.	*adoptar una actitud* *cobrar importancia* *iniciar una sesión* *tomar posición*	*take an attitude* *acquire importance* *start a session* *obtain a position*
$ContOper_1$ 16	Lat. *continuare* – 'to continue'. Continue to do, perform, experience, carry out K.	*guardar silencio* *mantener el equilibrio* *seguir un modelo* *llevar una vida (ocupada)*	*keep silence* *keep one's balance* *follow an example* *lead a (busy) life*

Table 7 presents LFs that we have chosen for our experiments. For each LF, we give the number of examples, its meaning, and sample verb-noun combinations.

In the lexical resource, we have annotated free word combinations with the tag FWC. The number of FWC is 261. We considered free word combinations as a lexical function FWC in its own right and experimented how machine-learning algorithms can predict this class of word combinations. Therefore, the total number of LFs we experimented with is nine.

Remember, that in the training set and test set, each verb-noun combination is represented as a set of all hypernyms of the noun and all hypernyms of the verb. To construct this representation, the number of sense for every verb and noun must be identified. However, sometimes, an appropriate sense was absent in the Spanish WordNet. Such words were tagged with abbreviation N/A (not available) instead of the number of word sense. In the training set, we included only verb-noun combinations that are disambiguated with word senses of the Spanish WordNet. In Table 8, the numbers of examples include only the verb-noun pairs in which all the words are disambiguated with the Spanish WordNet.

Table 8. Number of verb-noun combination in the test sets

Test set	Number of verb-noun combinations
100%	5181
75%	3886
50%	2590
25%	1295

The total number of examples for all 9 lexical functions is 900.

Training Sets

For each of 9 LF chosen for experiments, we built a training set, so we had 9 training sets. All training sets included the same list of 900 verb-noun combinations. The only difference between training sets was the annotation of examples as positive and negative. As an example, let us consider the training set for $Oper_1$. In the list of 900 verb-noun pairs, there are 266 examples of $Oper_1$, so these examples are marked as positive in the training set, and all the rest of verb-noun combinations whose number is 634 (900

– 266 = 634) were marked as negative examples. This procedure was applied to each training set.

Test Sets

The test sets were built independently of the training set. For this, 5181 verb-noun combinations for the test set were extracted from the Spanish Treebank Cast3LB [Civit and Martí 2004]. Four test sets were constructed, including, respectively, 100%, 75%, 50%, and 25% of all verb-noun pairs taken from Treebank Cast3LB. Words in the test set were not annotated with lexical functions. Table 9 gives the number of verb-noun pairs in all four test sets.

5 Conclusions

Lexical functions represent important linguistic information. Their understanding is important for correct interpretation of texts by a person or a computer. However, manual compilation of the corresponding dictionaries is a tedious and costly work. What is more, as any linguistic phenomenon, they depend on language, thematic domain, and genre, and they can change with time.

Automatic acquisition of lexical functions from unstructured raw texts greatly alleviates the problem of compilation of combinatorial dictionaries and in particular dictionaries of lexical functions. In this paper, we have presented a detailed discussion of the very notion of collocation and a review of existing approaches to automatic acquisition of lexical functions. Then we presented our methodology for representing the meaning of words via sets of direct and indirect hypernyms. Such a representation is particularly useful for automatic extraction of lexical functions from unprepared raw text corpora.

In our future work, we will apply this semantic representation to actual compilation of large dictionaries of lexical functions via supervised learning of semantic similarity measure between words.

References

1. [Alonso Ramos 2003] Alonso Ramos, M.: Hacia un Diccionario de colocaciones del español y su codificación. In: M. A. Martí et al. (eds.), Lexicografía computacional y semántica. Barcelona: Edicions de l'Universitat de Barcelona, pp. 11–34.
2. [Alonso Ramos et al. 2008] Alonso Ramos, M., Rambow O., Wanner L.: Using semantically annotated corpora to build collocation resources. Proceedings of LREC, Marrakesh, Morocco, pp. 1154–1158.
3. [Apresjan 2004] Apresjan, Ju. D.: About semantic nonemptiness and motivatedness of verbal lexical functions. (In Russian.) Voprosy jazykoznanija: pp. 3–18
4. [Benson 1986] Benson, M., Benson, E. and Ilson R. 1986. The BBI Combinatory Dictionary of English. John Benjamins, Amsterdam.
5. [Benson 1990] Benson, M. 1990. Collocations and general-purpose dictionaries. International Journal of Lexicography, 3(1), 23-35.

6. [Bolshakov and Gelbukh 1998] I. Bolshakov, A. Gelbukh. Lexical functions in Spanish. Proc. CIC-98, Simposium Internacional de Computación, November 11-13, 1998, Mexico D.F., pp. 383-395.
7. [Bolshakov and Gelbukh 2000] I.A. Bolshakov and A.F. Gelbukh. Classification of Collocations in a Lexical Database by Meaning of the Combined Words. In: Selected papers CIC-1999, CIC, IPN, Mexico City, 2000, pp. 5-15.
8. [Bolshakov and Gelbukh 2001] Igor A. Bolshakov and Alexander F. Gelbukh. A Large Database of Collocations and Semantic References: Interlingual Applications. International Journal of Translation, Vol.13, No.1-2, 2001, pp. 167-187.
9. [Bolshakov and Gelbukh 2002] Igor Bolshakov, Alexander Gelbukh. Word Combinations as an important part of modern electronic dictionaries. Procesamiento de Lenguaje Natural, No 29, 2002, Spain, p. 47-54.
10. [Bolshakov and Gelbukh 2004] I.A. Bolshakov, A. Gelbukh. Computational linguistics: models, resources, applications. IPN - UNAM - Fondo de Cultura Económica, Mexico, 187 pp.
11. [CEB] Concordancia electrónica de la Biblia Reina Valera 1960 online, http://www.concordancia.bravefire.com/concordancia.php/, last viewed on June 07, 2010
12. [Civit and Martí 2004] Civit, M., Martí, M.A.: Building Cast3LB: A Spanish Treebank. In: Research on Language and Computation, vol. 2(4), pp. 549–574. Springer, Netherlands
13. [Cowie 1994] Cowie, A. P. 1994. Phraseology. In Asher, R. E. (ed.). The Encyclopedia of Language and Linguistics. Oxford, Pergamon Press.
14. [DiCE] Diccionario de colocaciones del Español, http://www.dicesp.com/paginas/, last viewed June 08, 2010
15. [Firth 1957] Firth, J. R.: Modes of Meaning. In J. R. Firth, Papers in Linguistics 1934–1951 (pp. 190–215). Oxford: Oxford University Press.
16. [Fontecha 1941] Fontecha, C.: Glosario de voces comentadas en ediciones de textos clásicos. Madrid: CSIC
17. [Fontenelle 1994] Fontenelle, T. 1994. What on Earth are Collocations? English Today 10:4, 42-48.
18. [Gelbukh and Sidorov] A. Gelbukh, G. Sidorov. Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, Mexico, 2006, 240 pp.
19. [Gledhill 2000] Gledhill, C. J. 2000. Collocations in Science Writing. Tübingen, Gunten Narr Verlag.
20. [Halliday 1961] Halliday, M. A. K.: Categories of the Theory of Grammar. Word 17, 241–292.
21. [Hausmann 1984] Hausmann, F. J. 1984. Wortschatzlernen ist Kollokationslernen. Zum Lehren und Lernen französischer Wortverbindungen. Praxis des neusprachlichen Unterrichts, 31, pp. 395 – 406.
22. [Howarth 1996] Howarth, P. 1996. Phraseology in English academic writing. Some implications for language learning and dictionary making. Tübingen, Niemeyer.
23. [Kilgarriff et al. 2004] Kilgarriff, A., Rychly, P., Smrz, P. and Tugwell, D.: The Sketch Engine. In Proceedings of EURALEX. France, Université de Bretagne Sud: pp. 105–116
24. [McVey and Wegmann 2001] McVey Gill, M., Wegmann, B.: Streetwise Spanish Dictionary/Thesaurus. Chicago: McGraw-Hill.
25. [Mel'čuk 1996] Mel'čuk, I. 1996. Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In: Wanner, L. (Ed.) Lexical Functions in Lexicography and Natural Language Processing, pp.37-102. John Benjamin Publishing Company.

26. [Mel'čuk and Zholkovskij 1984] Mel'čuk, I. A. and Zholkovskij, A. K. 1984. An Explanatory Combinatorial Dictionary of the Contemporary Russian Language. Wiener Slawistischer Almanach, Sonderband 14, 1984.
27. [Mel'čuk et al. 1984] Igor Mel'čuk, Nadia Arbatchewsky-Jumarie, Lidija Iordanskaja, Adèle Lessard. Dictionnaire explicatif et combinatoire du français contemporain, Recherches lexico-sémantiques I. Les Presses de l'Université de Montréal, 1984.
28. [Mel'čuk et al. 1988] Mel'čuk, Igor, Nadia Arbatchewsky-Jumarie, Louise Dagenais, Léo Elnitsky, Lidija Iordanskaja, Marie-Noëlle Lefebvre, Suzanne Mantha. 1988. Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques II. Les Presses de l'Université de Montréal.
29. [Miller 1998] Miller, G.A: Foreword. In: Fellbaum, C. (ed.) WordNet. An Electronic Lexical Database, pp. xv–xxii. MIT Press, Cambridge, Mass. (1998)
30. [Nastase and Szpakowicz 2003] V. Nastase and S. Szpakowicz. Exploring noun-modifier semantic relations. In Fifth International Workshop on Computational Semantics (IWCS-5), Tilburg, The Netherlands, pages 285-301, 2003.
31. [Nastase et al. 2006] Nastase, V., J. Sayyad-Shiarabad,M. Sokolova, and S. Szpakowicz. 2006. Learning noun-modifier semantic relations with corpus-based and WordNet-based features. In Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference. AAAI Press.
32. [Oelschläger 1940] Oelschläger, V.R.B. 1940. A Medieval Spanish Word-List: A Preliminary Dated Vocabulary of First Appearances Up To Berceo. Madison, Wisc.: University of Wisconsin Press.
33. [RAE 2001] Real Academia Española 2001. Diccionario de la Lengua Española. (Twenty Second Edition.) Madrid: Real Academia Española
34. [Ruppenhofer et al. 2006] Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. R. and Scheffczyk, J. 2006. FrameNet II: Extended Theory and Practice. Available at http://framenet.icsi.berkeley.edu/book/book.pdf. ICSI Berkeley
35. [Sanromán 1998] B. Sanromán. Contribución lexicográfica al estudio de los nombres de emoción. Master's thesis, Universidad de Coruña.
36. [Sanromán 2003] B. Sanromán. Semántica, sintaxis y combinatoria léxica de los nombres de emoción en español. PhD thesis, Helsinki: University of Helsinki.
37. [Sinclair et al. 2004] J. Sinclair, S. Jones, R. Daley. English collocation studies: The OSTI report. Continuum. 2004.
38. [SpWC] Spanish Web Corpus in the Sketch Engine. 3 May 2010. http://trac.sketchengine.co.uk/wiki/Corpora/ SpanishWebCorpus
39. [SpWN] Spanish WordNet, http://www.lsi.upc.edu/~nlp/web/index.php?Itemid=57 &id=31& option=com_content&task=view, last viewed June 02, 2010
40. [Van Roey 1990] French-English Contrastive Lexicology: An Introduction. Louvain-la-Neuve, Peeters.
41. [Vossen 1998] P. Vossen. EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic, Dordrecht.
42. [Wanner 2004] Leo Wanner. Towards automatic fine-grained semantic classification of verb-noun collocations. Natural Language Engineering (2004), 10:2:95–143 Cambridge University Press.
43. [Wanner et al. 2006] Wanner, L., Bohnet, B. and Giereth, M. 2006. What is beyond Collocations? Insights from Machine Learning Experiments. EURALEX.

Ontology-based Semantic Relatedness Measures: Applications and Calculation

Alexander Gelbukh

Natural Language Processing Laboratory,
Centro de Investigación en Computación,
Instituto Politécnico Nacional.
07738 México D.F.
gelbukh@gelbukh.com

Abstract. We propose a procedure for measuring semantic relatedness of two words using an ontology, or semantic network dictionary. We discuss applications of this procedure in detail for lexical, syntactical, and co-reference disambiguation in natural language processing as well as in machine translation. In addition, we use a simplified version of this procedure for automatic translation of the semantic network itself into other languages. This simplifies creation and maintenance of semantic network dictionaries for different languages, thus enabling the described methods for processing of texts in languages other than English.

Keywords. Natural language processing, text processing, syntactic analysis, disambiguation, semantic network.

1 Introduction

Natural language processing is a branch of Artificial Intelligence and Computational Linguistics that studies methodologies and algorithms for automatic analysis of natural language, in the form of text or speech, and applications of linguistic processing of texts to many tasks important in practice.

Among various problems that arise in such automatic processing is the problem of ambiguity: the algorithm should make a choice between two or several possible variants of interpretation of the same linguistic unit, such as a word or syntactic dependency in a sentence. A promising way of resolving such ambiguities is to select the interpretation most consistent with the context. A specific notion of consistency is given by a so-called semantic relatedness measure: a numerical measure between the given unit (say, a word) and similar units located nearby in the text. Given a suitable semantic relatedness measure, the algorithm should measure the relatedness between all variants of interpretation of the given unit and all units in its vicinity and select the one that on average gives the best result (the strongest relationship).

Thus, the study of disambiguation methodologies can be largely reduced to the study of different definitions of semantic relatedness measures, and suitabil-

ity for a particular task, and their calculation basing on the available lexical resources. In this paper, we describe a particular semantic relatedness measure calculated using a semantic network dictionary such as WordNet [Miller, 1990] or FACTOTUM SemNet [Bolshakov et al., 1995b].

1.1 The curse of ambiguity

Probably the most difficult problem that nearly any algorithm dealing with the natural language faces is the curse of ambiguity. Be it just one word, or a phrase, or a text, very often there are several possible interpretations of what it means or what structure it has. We consider ambiguity resolution at all language levels the most important problem of natural language processing. To resolve the ambiguity, in much larger number of cases than it seems at the first glance, complicated reasoning or deep knowledge is necessary, often of semantic, pragmatic, or extra-linguistic nature.

A large number of works on ambiguity resolution employ manually crafted marked up text corpora, dictionaries [Luk, 1995], thesauri [Yarowsky, 1992], semantic networks [Sussna, 1993; Voorhees, 1993], or a combination of such lexical information sources [Yarowsky, 1995]. Still the problem is far from being satisfactorily solved.

In an ideal case, ambiguity resolution should be a side effect of some kind of "understanding," by which we mean construction of some detailed model of the whole situation described in the text and embedding it in the world model based on pre-existing knowledge, experience, or other texts read. The "true" linguistic knowledge, mostly lexical, ideally should be stored in vast dictionaries, such as combinatory dictionaries developed in frame of the Meaning ⇔ Text theory [Mel'cuk, 1974; Steel, 1990], or programmed in sophisticated procedures, such as in the Word Expert Parser model [Berleant and Daniel, 1995]. However, either manual or automatic compilation of such resources is extremely labor consuming and is hardly affordable in the nearest decades. On the other hand, such "true understanding" is too demanding computationally to be considered now; what is more, there seems to exist evidence that such a way is too computationally demanding even for human brain.

A way that is less computationally demanding is to use some pre-constructed pieces of "typical" situations and first of all to check the ambiguous constructions against them, addressing to a deeper analysis only when the choice cannot be made with simpler processing. Such pre-constructed pieces of information can be of different nature, such as syntagmatic, semantic, pragmatic, etc.

For instance, syntagmatic patterns could be represented by frequently used or "meaningful" word combinations, such as *take a bus*, *take a pen*, as opposed to **take weather* [Bolshakov *et al.*, 1995a]. Such a simplified set of syntagmatic patterns can be used (and probably is used by a human) in syntactic analysis instead of the much more expensive "true understanding."

In a similar manner, instead of a computationally demanding reasoning, a set of simplified "typical" semantic patterns can be used for disambiguation. Such

semantic patters could describe some atomic pieces of typical situations involving the words of the text. One of the form of representation of such knowledge is a semantic network, a set of semantic relationships between words in their specific senses.

1.2 Applications of semantic relatedness measures

This measure of relatedness is useful for resolving ambiguities of different types as well as for related tasks such as automatic translation of texts or even dictionaries. For instance, to resolve **syntactic** ambiguity, a variant of parsing should be chosen in which syntactically related words are more closely related semantically. To resolve **lexical** ambiguity between word senses in a context, the lexical variant should be chosen that is most closely related to the global or local topic of the document, or to the nearest words in the context [Banerjee and Pedersen, 2002; Patwardhan et al. 2003].

Similarly, to resolve **referential** ambiguity, the closest candidate is chosen to the words in the local context. In **text translation**, if the homonyms are not separated in the bilingual dictionary used for translation, the procedure of lexical disambiguation can be applied in the target language at the stage of text generation. Finally, in **translation of dictionaries** including the semantic network itself, lexical disambiguation can be performed on the reverse translation of the results back to the source language.

Various semantic relatedness measures have been proposed [Budanitsky and Hirst, 2001]; some of them are implemented in the freely available WordNet::Similarity package [Pedersen et al., 2004]. In this paper, we show how a semantic network dictionary can be used to measure the degree of semantic relatedness in a typical context between two given words [Gelbukh 1998].

The paper is organized as follows. In Section 2, we discuss a methodology for distance measurement in a semantic network. In Section 3, we present the applications of this methodology in various tasks related to computational linguistics and natural language processing. In Section 4, we discuss computational aspects of our methodology. Finally, Section 5 concludes the paper.

2 Distance measurements in a semantic network

In this section, we discuss the basic notions of measuring the semantic distances between two given words. The specific algorithms are given in the last section of the paper.

2.1 The structure of a semantic network dictionary

In our research, we used the FACTOTUM SemNet semantic network dictionary [Bolshakov et al., 1995b]. It is an English dictionary, though below we describe

how to use it for other languages, our target language being Spanish. There exist other semantic networks—most notably WordNet [Miller, 1990], which has been widely used for natural language processing because of its availability; a Spanish version is available in frame of European WordNet project. SemNet, however, has a larger number of types and a more flexible representation of semantic relationships, making it more suitable for natural language processing applications. However, our methodology can be applied to WordNet, too.

In its logical structure, the SemNet dictionary is a set of so-called *relationships* between pairs of concepts (in rare cases between sets, here we omit the corresponding details for simplicity).

In SemNet, a **concept** is usually a word, e.g., *book*, or a word combination, e.g., *address book*, referring to a specific thing or idea. In most cases textual words have several meanings; in this case they are marked with different numbers, e.g., *bill*$_1$ (banknote), versus *bill*$_2$ (check), *bill*$_3$ (declaration), *bill*$_4$ (pike), or *bill*$_5$ (ax). Often such word senses have different translations to other languages: in Spanish, *bill*$_1$ is *billete*, *bill*$_2$ is *cuenta*, *bill*$_3$ is *declaración*, *bill*$_4$ is *pico*, and *bill*$_5$ is *hacha*.

All such senses of any word, even closely related ones, have different identification numbers in the dictionary, are located at different positions, and often have different sets of relationships to other words; if needed, they can be connected with each other explicitly by a relationship. Thus, one word (character string) can represent different concepts.

In turn, one concept can be represented by several words. In this case, they are considered synonymous in these particular meanings, and are listed together in a *synset* to represent and disambiguate the concept, e.g., {*bill*$_1$, *note, banknote*}. Thus, generally by a concept we always mean a group of synonymous word senses. However, for convenience we name the concepts with just one of the words of the synset.

Relationships are used to connect two (or, rarely, more) concepts. They are labeled with different type names, such as IS_A, USES, CAUSES, etc. A relationship can be viewed as a simple statement expressing a "typical" fact, e.g., *computer IS_A equipment*, *explosion* CAUSES *damage*. There are some attributes, or properties, of individual relationships, like MAYBE, USUALLY, RARELY, etc., e.g., *seeing MAYBE* USES *telescope*. For human convenience, there are different ways to express the same fact in the SemNet dictionary, e.g., *telescope* IS_USED_FOR *seeing*, though all such expressions can be formally converted to a common internal representation. A fragment of such a network is shown in the Fig. 1. In this example, we can see that a telescope is a tool to see, an animal can have an object, etc.

In the human-readable form of the dictionary, the most extensive set of relationships—namely, most of the IS_A relationships—is represented implicitly by placing the concepts in hierarchical order in the dictionary. This does not imply that a concept may not be a subtype of several concepts, for example, a *girl* IS_A *child* and IS_A *female*; in this case, one of the relationships is indicated in the dictionary explicitly.

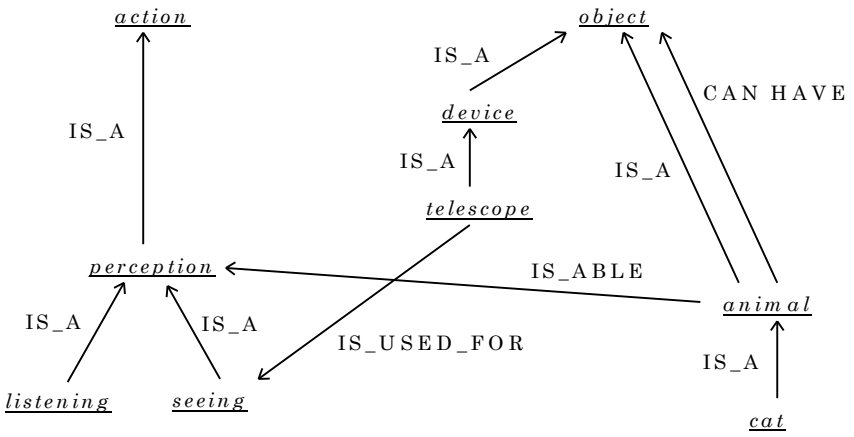

Fig. 1. A fragment of a semantic network.

Many possible relationships can be easily inferred by some general rules from other relationships; e.g., transitive relationships like IS_A and IS_PART_OF: if *a* IS_PART_OF *b* and *b* IS_PART_OF *c*, then *a* IS_PART_OF *c*. In such cases, only some of the relationships—the immediate ones—are explicitly included in the dictionary, to keep its size maintainable, e.g., *car* HAS_PART *motor* and *motor* HAS_PART *screw* implies *car* HAS_PART *screw*.

Other rules of inference involve particular relationships or groups of relationships. E.g., the most obvious one is that if *a IS_A b* and *b R c*, then *a R c*, where R is any relationship. In some cases, such inheritance of characteristics from higher categories is defeasible, i.e., it may be blocked explicitly by a special notation in the definition of a concept, or it may be canceled where contradictory information is inherited from more than one higher node.

2.2 Paths in the semantic network

The semantic network can be viewed as a graph. A **path** in such a graph is a chain of relationships r_1, ..., r_n such that r_i and r_{i+1} have exactly one common word, $i = 1, ..., n - 1$. If a word A is the beginning of the path and the word B is its end, we say that the path leads from A to B. There are several reasons to use paths for measuring the semantic closeness of words in the network.

First, since some of the relationships are present in the network only implicitly and can be inferred by application of the inference rules, a problem arises of generating all the relationships, including the implicit ones, between, say, two given words. The problem can be formulated as enumerating all the network paths with some special condition that lead from one given word to another.

Second, in some cases important commonality between words may not be expressed in terms of any existing named type of relationship. E.g., on Fig. 1, it

can be seen that "a *cat* CAN HAVE something that IS_USED_FOR *seeing*." There is no named type of relationship expressing this fact, so we have to represent the commonality between these two words by just a path of the two relationships.

Third, some rules of inference may have fuzzy character, being rather common-sense observations. Therefore, applying too many rules can make the result less reliable. In some cases, we can express this loss of reliability by adding the MAYBE attribute to the resulting inferred relationship, though in general case we should use some kind of lengths, or weights, of relationships and paths, as described below.

Finally, some participants of the situation described in the text may not be mentioned explicitly. E.g., in the phrase "*The seller asked the buyer for too high price*" there may be no explicit relationship in the dictionary between the words *seller* and *buyer*, though the relationship between them can be found through the implied actant, *goods*: *buyer* CONSUMES *goods* HAS_SOURCE *seller*, a path of two relationships.

2.3 Lengths of the relationships and paths

In general, we need to assign some weight, or "length," to each relationship and calculate the "length" of a path based not just on the number of links in it, but also on their individual lengths. This value gives the quantitative estimation of how closely related are the two given words, while the path itself with the labeled links gives the qualitative estimation of exactly how the two given words are related.

For **explicit** relationships, such a length can reflect the degree of the importance of the relationship, e.g., IS_A relationships indicate that the words are close and probably substitutable for each other in most contexts. On the other hand, CONSUMES relationship reflects much less degree of closeness, i.e., it is "longer."

For **inferred** relationships, their lengths can depend on the kind of relationships involved in the logical inference (i.e., be an attribute of the corresponding rule) and on the length of the logical chain. E.g., many applications of the transitivity rule for IS_A relationships can increase the length of the resulting IS_A relationship. In Fig. 1, it is true but "less reliable" that *cat* CAN HAVE *telescope*, due to too many applications of IS_A transitivity rule. The fuzzy character of the inference rules is obvious for such relationships as IS_SIMILAR_TO, which is "to some degree" transitive.

For a **path**, its length should increase with the total length of the constituting links. The longer the path between two words, the less semantically related they are.

An easy way to assign the lengths to the links is to relate a specific length value to each **type** of the links, e.g., 1 to IS_A, 5 to SIMILAR_TO, and 20 to CAUSED_BY. Such assignment may be context-dependent or may vary according to the type of information being retrieved from the text. E.g., in a text

with the principal topic "toys" the relationship SIMILAR_APPEARANCE can be more important than in a text with another principal topic. We leave such considerations for the future.

Sometimes it might be desirable to assign a length to **individual** links or groups of links. Ideally, each individual link should have some specific length expressing the degree of commonality between the two specific words. Assignment of these values hardly can be done by hand; instead, some procedure for training on a large corpus might be used in the future. There are cases, though, when additional individual coefficients can be assigned automatically.

For instance, special precautions are to be taken in order to prevent the algorithm from abuse of hierarchical links like IS_A. Namely, any concept referring to an object IS_A *object*: *car* IS_A *object*, *book* IS_A *object*. Thus, any two objects are connected with a path of two (usually implicit) links: *car* IS_A *object* HAS_SUBTYPE *book*, which normally should not imply a great degree of commonality between them. On the other hand, in some cases a path of two IS_A relationships does imply commonality: *Ford* IS_A *car* HAS_SUBTYPE *BMW*.

This problem is already mitigated by assigning to the implicit relationship a greater length (corresponding to a weaker relationship) than the length of an explicit relationship, when the implicit relationship is computed by application of the inference rules. However, the precision of the procedure can be improved by assigning the greater length to the hierarchical links located near the top of the hierarchy, thus, the length of the link *thing* IS_A *object* is more than that of the link *Ford* IS_A *car*. Namely, on the stage of preparation of the relationships database, the *maximum* number of links is determined from each node to the top of the hierarchy, and the links leading to this node are scaled correspondingly.

For example, in Fig. 2, the distance between *Ford* and *linguist* is much greater than between *Ford* and *BMW*, though both ones are subtypes of *object*. The distance between *car* and *book* is greater than the distance between *Ford* and *BMW*, though in both pairs there are exactly two explicit links between the nodes.

2.4 Shortest paths problem

To determine the semantic distance between two words, the shortest possible path in the network is to be found; its length can be used as an estimation of the degree of their semantic nearness. Since not all paths can be acceptable in a specific context, in some, and supposedly many, cases the next, then next, etc., shortest path should be used to determine the measure of the semantic distance between two words in a specific *context*. E.g., if the context suggests that the possible relationship between the two words is USES (expressed by the preposition *with*), then even a shorter relationship of the type IS_A cannot be used as a measure of closeness between these words in this context.

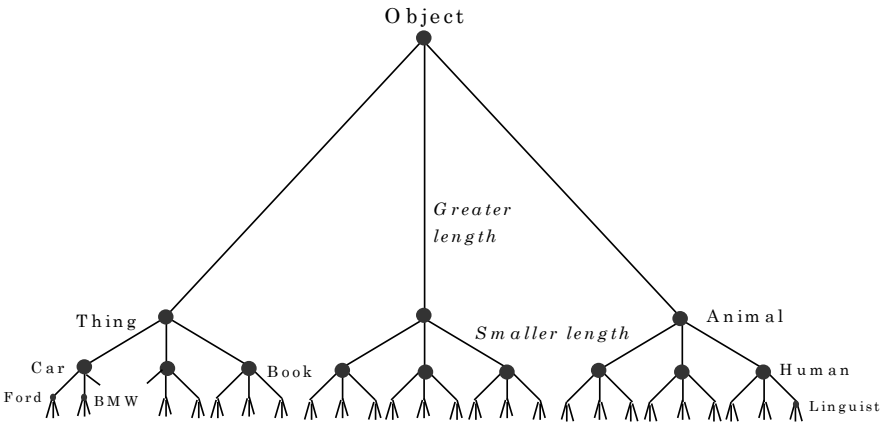

Fig. 2. Different lengths of hierarchical links.

The curse of ambiguity manifests itself in the full degree in the task of finding such paths. There is virtually infinite number of paths in the network, connecting the two given words. The computational aspects of the problem of finding the shortest path are discussed in the last section of the article; here it is enough to mention that the problem has well-known solutions and the only mathematical issue is computational efficiency. Thus, we will first discuss the linguistic applications of such an algorithm, assuming that it operates on a large enough semantic network dictionary.

3 Applications in computational linguistics tasks

Finding the shortest paths in the semantic network between two given concepts and measuring their relatedness in the network in a specific context has numerous applications for disambiguation in language processing and automatic translation. Note that in practice one can adopt a methodology where different sources of evidence and measures of semantic relatedness "vote" for the final decision; here we propose one of such "voters."

3.1 Syntactical disambiguation

Consider a phrase "*John sees a cat with a telescope*." The phrase is syntactically ambiguous: does it mean 'John uses a telescope to see a cat' or 'John sees a cat that has a telescope,' or 'John sees a cat and a telescope,' or maybe 'John that has a telescope sees a cat,' etc.? This ambiguity cannot be resolved using only lexical or syntactical information, since all the interpretations are syntactically plausible.

Most methods employed currently for solving this ambiguity, such as probabilistic grammars, rely on supervised machine learning to learn probabilities of

different syntactic links, or, in the case of lexicalized grammars, the probabilities of combining specific words. With this information, a parse variant that contains most probable links is preferred to other variants and is chosen as the output of the parser.

While such methods give excellent results, they have certain disadvantages. The first disadvantage that we can mention here is the need in manually marked up corpora, called treebanks. Such corpora are expensive in development, and they do not yet exist for all languages; in fact, such corpora of considerable size exist only for a few major languages.

Another important disadvantage of statistical methods for this task is the data sparseness effect: while such training corpora have plenty of examples for frequent phenomena, due to the Zipf distribution law they lack a reliable number of examples for less frequent cases. In contrast, manually crafted linguistic resources tend to pay attention to linguistic phenomena irrespectively of their frequency, and thus provide information for both frequent and infrequent usage cases.

Therefore, in this paper we will assume so-called symbolic approach, in contrast to more widespread statistical approach. The symbolic approach relies on manually crafted dictionaries and grammars. In particular, it allows for exploiting existing lexical resources and dictionaries, including those created in pre-computer era for the use of human readers and not automatic procedures. The dictionary we used for this work, FACTOTUM SemNet, is based on the classical Roget thesaurus, which to some degree guarantees high quality of the information it contains.

In Fig. 3, the first two of abovementioned variants for the analysis of the phrase "*John sees a cat with a telescope*" are presented. The syntactic dependencies in question are *see* → *telescope* and *cat* → *telescope*; what are the semantic relationships between these words? There is a relatively short path between *seeing* and *telescope* in the semantic network dictionary. What is more, we can note that the type of the relationship(s) constituting this path agrees with the supposed instrumental syntactic relationship between these words in the phrase.

On the other hand, the best path between any sense of *cat* and *telescope* that agrees with the type of the supposed syntactic dependency is much longer. Thus, the variant (1) should be chosen here. This, though, should not prevent the linguistic processor from being able to backtrack and revise this decision later if the subsequent sentences disagree with this choice.

Sometimes just the quantitative measure of the nearness (the weighted length of the path) can be used for comparison. However, for better quality of analysis the whole path should be checked against the expected syntactical type of the relationship. E.g., in a phrase "*John sees a cat with a boy"* there is a short path between *seeing* and *boy*: *boy* IS_ABLE *see*, but the type of the relationship contradicts with the hypothesis that *boy* here is a tool to *see* with. This is why the procedure for finding the paths should be able to enumerate the paths until an acceptable one is found.

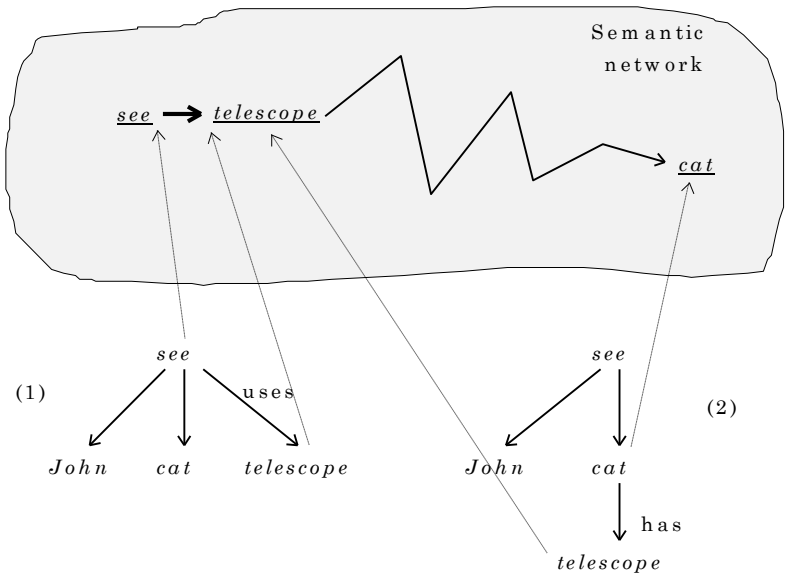

Fig. 3. Example of syntactic disambiguation process.

3.2 Lexical disambiguation

Ambiguity also arises in selection of a particular sense of a word in a phrase. Sometimes they can be resolved at syntactic level, usually when the choice is made between different parts of speech, e.g., in the phrase "*John tables the dishes"* the word *tables* is clearly a verb. However, in many cases, especially when a word has different meanings within the same part of speech, semantic information has to be employed.

Compare, for example, the phrases "*There were fruits and drinks on the table*" and "*The numbers were arranged in a table*." By addressing a semantic network, it can be determined that in the first phrase the shortest path exists between other words and the sense 'table as a furniture,' while in the second phrase, the shortest path leads from *numbers* to 'table as a picture.'

It is not as clear as it is with syntactic ambiguity, with what words in the phrase the given word is to be compared. Good candidates can be words close in the syntactical structure to the given one. Other good candidates are the words describing the main global or local topics of the document. For example, if the document in general is on mathematics, the word *table* will likely be used in it as 'table as a picture,' even if the nearest words do not suggest this directly.

Local and global topics of the document can be determined with the approach called CLASITEX [Guzmán-Arenas, 1997]. In this approach, all the textual words of the document without any preliminary disambiguation are clustered with the help of a semantic network dictionary. The centers of the largest clusters represent the main topics of the document. The noise is canceled out since

the wrong senses of the words form smaller clusters. This approach can be applied to a part of the document, revealing the local topics. These topics can also be used in the disambiguation process: the distance is to be measured between them and the current word being disambiguated.

Since in the process of disambiguation, many words or word senses (global and local topics, surrounding words, etc.) possibly have to be tried and the results have to be accumulated, the procedure may be computationally demanding. However, in comparison with, say, Word Expert Parser model [Small and Rieger, 1982], our procedure requires easier available data and can be used in frame of the traditional text processing algorithms.

3.3 Referential disambiguation

The problem of referential disambiguation arises each time a pronoun, ellipsis, or zero subject (very common in such languages as Spanish) is used in the text. In general, at the stage of text analysis such a reference must be replaced with another word probably used somewhere in the text. Though there are linguistic considerations on selecting the candidates to fill the valence, they usually give ambiguous results when only lexical and syntactic information is considered.

However, it is possible to resolve this task into the task of lexical disambiguation. Namely, when several candidates are to be tried to fill the valence, they can be just treated as different "senses" of the pronoun in this particular context. Then the procedure described in the previous section can be applied with nearly no modifications. The only difference is that neither global nor local topics are used in the comparison.

3.4 Machine translation

In general, text translation is a quite different task from text understanding. Ideally, translation should include the steps of text understanding in the source language and then text generating in the target language. If the ambiguity is resolved at the stage of analysis, and if the bilingual dictionary is good enough, there should be no problems with ambiguity during text generation. However, in real life it is not the case, for both practical and theoretical reasons [Narin'yani, 1997].

In practice, less sophisticated methods are currently used, working mostly at the syntactic level. Some of commercially available symbolic-based translation systems distinguish the senses of the words only by a limited number of semantic classes or by literal recognition of some number of idioms. E.g., this phrase was translated from Spanish by Globalink's Power Translator Professional: "*El artista realiza bien el papel*" ⇒ "*The artist accomplishes well the paper*" (instead of *role*). This program, though, does distinguish these senses in some contexts: it seems to make choice based on literal recognition of the idiom "*jugar un papel*," e.g., "*El diputado juega un papel importante*" gives "*The deputy*

plays a role important," but: "*El diputado juega el papel más importante*" gives "*The deputy plays the most important paper*." For good symbolic-based automatic translation, there must be available (1) a good disambiguation procedure in the source language, (2) a good bilingual dictionary that translates one-to-one senses to senses, not textual words to sets of words. Both conditions are very difficult to satisfy. For example, there might not be available a Spanish dictionary to disambiguate the two senses of the word *papel*.

In addition, the most elaborated up to date dictionaries, including academic dictionaries, usually provide translations of a word into several possible words in the target language, e.g.: "*papel*: paper; document; role; <...>" [Spanish-English, 1963]. In this case, even if the senses have been disambiguated in the source language, the dictionary anyway does not contain the information necessary to translate them one-to-one into words of the target language.

Our methodology permits to disambiguate the words after translation in the target language. As in the previous section, we can treat the ambiguous position as a word with several "senses" and then apply the procedure of lexical disambiguation to the generated phrase in the target language.

For instance, in the example above, there is a shorter path in the English semantic network between *artist* and *role* than between *artist* and *paper* or *document*. This allows us to use a semantic network to improve the results of translation made with existing bilingual dictionaries, rather than developing new sense-to-sense dictionaries, which are expensive to create and difficult to share between different systems due to their tight integration with the other modules of a linguistic processor.

3.5 Automatic translation of the semantic network

Our disambiguation procedure can be applied to automatic or semi-automatic translation of the semantic network itself into other languages. Since we have taken part in such a translation project (though the work was mainly done by hand), we are aware of all the deficiencies of the very idea of translation of a semantic network, and of low quality of the resulting dictionary [Bolshakov *et al.*, 1995b]. Still there are at least three reasons to translate semantic networks.

First, creating a semantic network from zero is a very difficult and expensive work. If the way the results are used is tolerant to the incompleteness and minor inaccuracies, it may be more efficient to use a lower quality dictionary translated from an existing resource than to wait for a better dictionary to be created in the far future. Actually, we believe that due to the nature of the functioning of natural language, any language processing software must be tolerant enough to incomplete and inaccurate information. However, since the semantic network contains mostly the facts about real-world objects and ideas, and in part due to commonality between the languages, most of the relationships tend to be translated correctly (though this may depend on the languages and subject area).

Second, the linguistic resources for such languages as English, French, Japanese, etc., are maintained by many people and groups all over the world, with

much money spent on their development, enlargement, and refinement. It would be a waste of effort to repeat all this work in full size for each language. Thus for groups that work on, say, Spanish language, to take advantage of the efforts spent in the world on development of English semantic networks, it is necessary not only to translate the first draft of the dictionary from English, but to be able to repeat such translation automatically as new versions of the English dictionaries become available. There is no need to mention that the existing machine translation programs designed for translation of phrases in a discourse are not appropriate to translate structured resources such as dictionaries; thus the necessity to create specialized dictionary translation software.

Third, existing ontologies can be used, such as so-called T2 Reference Ontology for English. Automatic or semi-automatic procedure for translation of this resource can be very useful in maintaining compatibility between semantic networks in different languages and the ANSI standard.

Details of the translation procedure are beyond the scope of this paper. Here we only discuss the application of the procedure for enumerating the paths in the network between two given points to the task of translation of the semantic network itself. However, automatic translation of a semantic network faces the same main problem: ambiguity. Each word in each its occurrence in the text of the dictionary, presumably in different senses, is translated by an ordinary bilingual dictionary to several different words of the target language.

The following procedure is proposed to choose the correct variant of the translation, using the same (English in our case) semantic network. Each variant of translation of a word is translated back to the source language. Then the distance in the source semantic network is measured between the source word and each variant of such a reverse translation. The variant(s) of translation are chosen, at least one of whose reverse translations is located near the source word sense in the network, i.e., there is a "short" enough path from this variant to the source word sense, see Fig. 4.

The copy of the source word is removed from the set of the reverse translations. Words having only one reverse translation, namely the same source word, are treated as special cases. They are inserted in the resulting dictionary, and if the source word has different senses, such words are marked when automatically inserted in the dictionary, and then checked by hand.

In theory, only the words with a reverse translation within the same concept, i.e., at the zero distance from the source word, should be accepted. However, in practice, a bilingual dictionary in most cases does not gives such accurate results; therefore, the paths of nonzero length should be taken into account.

For the set of the paths in the network to be considered, for each textual word all its senses should be tried unless any disambiguating information is available in the bilingual dictionary; usually it is not. The choice of the candidate is made in two steps. First, the weights, i.e., the lengths of the corresponding paths, of the reverse translations of each candidate are combined to calculate the weight of the candidate itself. Second, the candidate(s) are chosen with the best such weight.

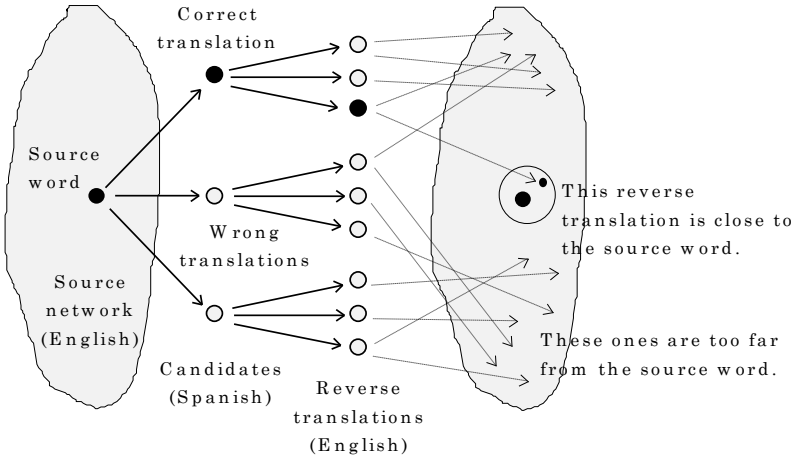

Fig. 4. Translation of a semantic network (two copies of the same network are shown to simplify the picture)

Various procedures can be used for both calculations. To combine the weights of the reverse translations for one candidate, in the simplest case a maximum (but not average) can be taken. In a more sophisticated procedure, the values for all the reverse translations better than some threshold should be accumulated. To choose the acceptable candidates of translation, in the simplest case only the best one is taken for each word, or all the candidates are accepted that are better than some threshold value. More sophisticated procedures can also be tried. For example, all candidates better than some threshold value should be accepted, all candidates worse than some threshold value should be rejected, or the best one should be chosen from those candidates whose weight falls between these two thresholds. The obtained semantic network dictionary may be then post-edited by hand. To be able to repeat the translation as new versions of the source dictionary become available, the changes made by hand should be saved in a special protocol.

As compared with the procedure of enumerating the paths used for text processing, the procedure used for translation of the dictionary itself can be simplified by ignoring completely the inference rules, since in this case the meanings should be preserved much more precisely. The length of the path can be calculated as just the number of links in it. This makes the implementation of the procedure for translation much more straightforward than for text understanding.

Inference rules can be used for better results. However, application of each rule should substantially increment the length of the path. E.g., a chain of transitive relationships like IS_A should be considered long enough, whereas the procedure used for text understanding would use the length near to 1 for such a path. The choice here, as well as the selection of the thresholds mentioned above, is made on the basis of desired compromise between the accuracy of the

translation (less usage of inference, higher thresholds) and the number of words that will get any translation at all (more usage of inference, lower thresholds).

Only basic relationships, such as IS_A and possibly IS_PART_OF and a few other, should appear in the paths, but not such relationships as USES, etc.

4 Computational aspects

Here we describe the mathematical problem statement and possible algorithms for its solution. Generally, a simple modification of Dijkstra's shortest path-finding algorithm [Dijkstra, 1959] could suffice, though we present a more sophisticated modification, adapted to large sparse networks.

4.1 Problem statement

Finding paths in the network is important for computational linguistic applications, primarily to measure the distance in the network between the two given words in a specific context. Usually to measure such a distance in a specific context, the shortest paths between the two points are to be found; however, it is not always true that necessarily the very short, optimal, path should be found first. There are several reasons for this.

First, there are many **restrictions** on suitability of particular types of paths in a specific context. If, for example, the syntactic relationship between the two given words suggests, say, instrumental relationship, then the semantic relationships of any other types, even very short, are useless in this context. Such restrictions can be applied after the path is found in the network. Thus, it is probable that the very short path will prove unusable in a specific context; in this case, the next path should be retrieved and examined. Thus, there is no point to apply a computationally demanding procedure to optimize completely the search process.

Second, the rules for calculating the length of a path in a specific context can be **context-dependent** and complicated. They may be better applied after the path has been found. This problem is discussed in more detail below at the end of this section.

Third, the **low precision** of all language data, including in the first place the text itself being analyzed, but also the dictionaries, grammars, etc., makes very precise procedures not so necessary. Everything in language is vague; any text is full with hints, omissions, implied information, metaphors, rather than being a collection of clear and simple logical statements. This makes too precise procedures of text processing in many cases useless.

On the other hand, since automatic procedures are applied to huge amounts of texts, **performance** is important, as long as the result of the analysis fits in the same confidence interval. Performance is especially important since the procedure for semantic distance measurement is invoked very many times in typical applications, such as referential disambiguation.

Therefore, we can consider the following problem: to enumerate the paths in a network between two given nodes, as a tendency starting from shorter ones, under the following conditions:

- Various timeouts apply, e.g., a threshold on the length of the paths: only the paths of the length less than a threshold value are considered.
- Computational effectiveness is a priority.
- Accuracy is the second priority. Better paths should go first, but only on average.
- The network is large and stored in a database, so that retrieval of the links leading from a given point is the most time-consuming operation.

At each step, we estimate—again, with some probability—the lower limit of the length of the paths the procedure can find yet. The importance of this latter requirement will be discussed below. We suppose that the calling routine at some will moment stop the enumerating process, or some kind of time-out is used to prevent the algorithm from infinite work, such as a restriction on the number of paths, or on their lengths, so that if the procedure cannot find any paths shorter than some threshold, it should stop. In addition, some qualitative restrictions may be imposed on the desirable paths, e.g., not to contain a particular relationship.

The problem is very similar to the well-known problem of finding the shortest paths in sparse graphs, e.g., [Shier, 1976; Johnson, 1977; Minieka, 1978]. However, there are some differences in the goals and conditions with the classical problems of optimization. The main differences between the two problems are summarized in Table 1.

Table 1. Comparison of the considered problem and the classical one.

Classical Optimization	Our problem
There are no restrictions on the length of the path.	Only the paths shorter than some threshold value (i.e., short enough ones) are considered.
Only one path is searched for (in some variants K paths).	Paths must be enumerated until the caller "accepts" one.
The very best path must be found.	Better paths should generally go first, but not necessarily the very best one is to be the first.
The length of a path is a mere sum of the lengths, or weights, of the individual links.	The length is calculated according to the fuzzy rules of combination of the relationships.
No previously prepared data is usually used.	Some data can be prepared in the database in advance.
The graph is small enough to be kept in memory.	The graph is very big and is stored on the hard disk.

By better paths, in Table 1 we mean the ones with smaller length, which usually means ones that contain fewer links or links with smaller lengths. This measure is computed as a combination of the lengths of individual links, with application of, or taking into account, the rules of inference. E.g., a chain of five IS_A relationships may be considered "shorter" than a chain of two IS_SIMILAR_TO relationships. In general, such an estimation is a complex problem by itself, and we will not describe it here in more detail.

The measure of length used by the algorithm can differ from the measure that is used by the calling procedure, the latter being probably calculated or refined by the caller itself with the application of some specific, possibly complex, rules, for example, making use of the logical structure of the situation described in the text itself.

This difference arises from our intention to separate the information internal to the semantic network from the information used in various applications, and to provide a general procedure (probably implemented as a separate module) that permits the caller to treat the semantic network as a black box. However, some minimal adjustments of the procedure will anyway be necessary for some applications; they are discussed below in the sections devoted to the corresponding applications.

We assume, therefore, that the algorithm should find the paths just good in some general meaning, and the caller will check if the path is in fact good for it, though the "generally good" paths should be usually good enough for the caller.

Therefore, the algorithm should not even try to optimize completely the enumerating process, since anyway chances are little that the very best in general sense path will be the very best for the caller, and we expect usually it will not. This changes the approach to the algorithm as compared with the classical optimization problems.

4.2 Algorithms

While there is extensive research devoted to the shortest paths problem, we are not aware of any known algorithm for solving exactly our task. It is not our goal in this paper to propose a mathematically refined algorithm, since at the current stage of the research we are mostly interested in the linguistic applications of the idea itself. However, we describe here some variants of the algorithms we currently use.

4.2.1 The case of equal lengths of the links

Here we consider a non-weighted graph. A simple algorithm of enumerating all the paths, a modification of Dijksatra's one, is as follows. Define a sphere S_r (A) around the point A as the set, actually a tree, of all the paths of the length *r* leading from the point A. (Each path in the tree is compactly represented by the ending point, additional characteristics such as the length, and a pointer to the previous path in the tree. When a new path is formed by adding one link to

the previous one, only such a data structure is to be created in memory.) Since we consider here the length of each link in the network to be just 1, the radii of the spheres are natural numbers; we can also consider S_0 (A) being the empty path, i.e., the point A itself. We call the set of ends of all the paths of the sphere S (A) its surface.

The next sphere S_{i+1} (A) can be formed by adding to S_i (A) each link leading from each its surface point. If necessary, obvious precautions can be taken to prevent the paths being formed from cycles, at least from the cycles formed by two copies of the same link passed in the opposite directions; this can be done by comparison of the link being added to a path with the immediately previous link in the path. Other types of cycles in a sparse network usually do not present much problems for our task.

The algorithm alternates between increasing the spheres S (A) and S (B), starting from, say, S (A). At each step, the intersection of the *surfaces* of the spheres S_i (A) and S_j (B), $j = i$ or $j = i - 1$, gives the paths of the length $i + j$. This algorithm enumerates all the paths between A and B, starting from the shortest ones.

In case of an oriented graph, when only the oriented paths from A to B are to be found, a simple modification of this algorithm can be used. The sphere S^+_{i+1} (A) should be formed only with the links leading *from* the points of S^+_i (A), while the sphere S^-_{i+1} (B) should be formed only with the links leading *to* the points of S^-_i (B). If the paths both from A to B and from B to A are to be found, two spheres S^+ and S^- are maintained for A and B, consisting of the links leading to and from the points, correspondingly. If there are other restrictions on the types of the paths, they also can be taken into consideration at the step of increasing the spheres.

4.2.2 Different lengths and inference rules

The algorithm described in the previous section can be generalized to the case of weighted graphs. We consider here a modification that not always gives the shortest paths first, but does so as a tendency. This algorithm can be easily modified to enumerate the paths in the proper order, but with slightly lower efficiency.

In this algorithm, the sets of paths, which we will still call spheres, actually are not spheres, i.e., the paths in such "spheres" do not have the same lengths. We define these spheres just recursively, the sphere S_{i+1} (A) being formed by adding to S_i (A) all the links leading from some of its surface points (we chose to add all the links here since the operation of retrieval of the links is the most time-consuming). In this case, not all the surface points of S_i (A) are expanded, instead, expanded are only the paths, usually one path, with the minimal length among all the paths of S_i (A). The surface of S_{i+1} (A) is defined by replacing the expanded points of the surface of S_i (A) with the ends of the newly added links.

Similarly, the spheres S (A) and S (B) are increased in turn, and the intersection of their surfaces gives the different paths between A and B. It is easy to

prove that this algorithm enumerates all the paths. Namely, let us call the minimal length of a path in the sphere its *minimal radius*. Each step of the algorithm increases the minimal radius of one sphere, and if the current minimal radii of S (A) and S (B) are r_1 and r_2, then all the paths with the lengths of $r_1 + r_2$ have been already enumerated by this moment.

Our algorithm does not enumerate the paths in the exact order of their length. A counter-example can be constructed when two points are connected by two long links (thus the length of the path is large) and, in addition, are connected by three short links (so that the length of the resulting path is small). In this case, the algorithm finds the former path first, while the latter path is shorter.

However, generally it tends to enumerate the shorter paths before the longer ones. It is possible to store the found paths temporarily without reporting them to the output, until the value $r_1 + r_2$ reaches the length of a temporarily stored path. With this modification, the algorithm will enumerate the paths in the proper order. However, for our goals we chose to use the path as soon as the algorithm finds it.

The inference rules and the rules for determining the lengths of different combinations of the relationships can be taken into account at the step of increasing the spheres and at the step of determining the intersection of the spheres. Namely, when a link is added to the path, the length of the path being formed is determined accordingly to the rules of combination of links. Similarly, when a complete path between A and B is formed by connecting together two paths, one of S (A) and another of S (B), the length of the combination may differ from the sum of the length of the two paths. This contributes in the lack of order in enumeration process. However, the shorter paths still tend to be enumerated before the longer ones.

4.2.3. Use of pre-calculated data

If the network is not a small world graph, the methods described above are good only to find short enough paths, since spheres of big radii are too large. In practice, it may not be a problem if only short paths are important for the applications.

However, if longer paths are required, an additional network of "pivot nodes" with pre-calculated information about their connections with each other may be used. This is similar to the idea of a cellular telephone system, where two phones, instead of communicating with each other, communicate with nearby nodes of a dense enough network, while those nodes can then communicate with each other in a predefined manner.

For this, at the stage of preparation of the database, nodes are added, or existing nodes are used, at nearly equal distances from each other and not further than some threshold distance from any node in the network. The number of such control nodes should be much less than the total number of nodes in the network. Information is stored with those nodes to help finding the paths leading from each of them to each another. To find a path from an arbitrary point A

in the network to another point B, first, the paths from each one of these points to the nearest pivot node are determined using the method of increasing spheres described above. Then, the path between the control nodes is retrieved from the database. Finally, the complete resulting path can be varied or optimized locally around the retrieved one.

4.2.4. Multiple comparisons

Sometimes not only the distance between two given nodes is to be determined. Instead, the questions to be answered are as follows: (1) which point in some set of points is the nearest to a given one, or (2) what are the two nearest points in a two given sets. These problems arise in disambiguation of the binding of a prepositional phrase and in referential disambiguation, correspondingly.

A simple modification of our algorithm allows us to take advantage of alternating between increasing the spheres in turn and of using the same sphere to determine the distances from the given point A to each of the points B_1, ..., B_n. All the spheres are increased, each one in its turn. Suppose we find at least one path between A and, say, B_1 such that its length is smaller than the sum of the minimal radii of the spheres S (A) and S (B_2), and no shorter paths have yet been found between A and B_2. Then the distance between A and B_1 is smaller than the distance between A and B_2. Note that since the paths are retrieved not in a completely precise order, the check against the minimal radii is important.

Given the complicated rules of link combination, the latter criterion is not precise, since the length of a path can be different from the sum of the lengths of its parts. Currently, we ignore this complication, since we consider the loss of precision to be less than normal fuzziness of all data related to natural language. In case of serious problems arising in understanding of a particular phrase, backtracking can be used later to calculate the distances in question more precisely.

5 Conclusions

A number of semantic network dictionaries and ontologies are available nowadays, mostly for English language, such as WordNet or the FACTOTUM SemNet dictionary.

We have presented in this paper a simple procedure, namely the search for the shortest paths in a sparse network, that can be used for determining the measure of semantic relatedness of two given word senses in a very large semantic network. This measure is useful for disambiguation in a variety of important tasks of natural language processing, such as lexical, syntactic, and referential disambiguation, as well as in text generation and machine translation.

In addition, this procedure can be used to translate automatically the semantic network dictionary itself into other languages. This makes our methods usable for processing of languages other than English. This also simplifies creation and

maintenance of semantic network dictionaries for these languages. What is more, such automatic translation of a semantic network will be useful in development and maintenance of semantic networks in languages other than English, which would conform to the ANSI Standard Ontology (T2).

In our future work, we plan to consider more detailed information for syntactic disambiguation that can be extracted from existing dictionaries [Castro-Sánchez and Sidorov, 2012]. We also plan to combine methods for improving the translation results developed in this paper with statistical methods based on alignment of parallel bilingual text corpora [Sidorov et al., 2011].

Acknowledgments

The work was partially supported by CONACYT grant 50206-H, SNI, and FP7-PEOPLE-2010-IRSES: Web Information Quality – Evaluation Initiative (WIQ-EI) European Commission project 269180.

References

1. [Banerjee and Pedersen, 2002] Banerjee, Satanjeev, and Ted Pedersen. *An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet*. In: Proceedings of CICLing 2002, the Third International Conference on Intelligent Text Processing and Computational Linguistics, pp. 136–145, 2002, Mexico City.
2. [Berleant and Daniel, 1995] Berleant, Daniel. *Engineering "word experts" for word sense disambiguation.* In Natural Language Engineering 1, 1995: pp. 339-362.
3. [Bolshakov *et al.*, 1995a] Bolshakov, I.A., P.J.Cassidy, A.F.Gelbukh. *CrossLexica -- a dictionary of collocations and thesaurus of the general Russian lexicon* (in Russian, abstract in English). In Proceedings of International Workshop Dialogue'95: Computational Linguistics and its Applications, Khazan, 1995.
4. [Bolshakov *et al.*, 1995b] Bolshakov, I.A., P.J. Cassidy, A.F. Gelbukh. *Parallel English and Russian hierarchical thesauri with semantic links, based on an enriched Roget's thesaurus* (in Russian, abstract in English). In Proceedings of International Workshop Dialogue'95: Computational Linguistics and its Applications, Khazan, 1995.
5. [Budanitsky and Hirst, 2001] Budanitsky, Alexander and Graeme Hirst. *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. In: Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, June 2001.
6. [Castro-Sánchez and Sidorov, 2012] Castro-Sánchez, Noé Alejandro and Grigori Sidorov. *Extracción automática de los patrones de rección de verbos de los diccionarios explicativos*. Polibits, vol. 45, 2012, pp. 67–74.
7. [Dijkstra, 1959] Dijkstra, E.W. *A note of two problems in connection with graphs*. Numerische Matematik, 1959, V. 1, pp. 269-271.
8. [Gelbukh, 1997] Gelbukh, A. *Using a Semantic Network for Lexical and Syntactic Disambiguation*. In Proc. of the International Symposium "CIC-97: Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación," November 12-14, 1997, Mexico D.F.

9. [Gelbukh, 1998] Gelbukh, A. *Using a Semantic Network Dictionary in Some Tasks of Disambiguation and Translation*. Technical report, Serie Roja, N 36. CIC, IPN, 1998.
10. [Guzmán-Arenas, 1997] Guzmán-Arenas, A. *Determining principal themes in a Spanish article* (in Spanish). In Proc. of the International Symposium "CIC-97: Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación," November 12-14, 1997, Mexico D.F.
11. [Johnson, 1977] Johnson, D.B. *Efficient algorithms for shortest paths in sparse networks*. J. ACM, 1997. Vol. 24, N 1, pp. 1-13.
12. [Luk, 1995] Alpha K. Luk, *Statistical Sense Disambiguation with Relatively Small Corpora Using Dictionary Definitions*. In Proceedings of the 33rd Annual Meeting of the Amer. Soc. for Comp. Ling., 1995, pp. 181–188.
13. [Mel'cuk, 1974] Mel'cuk, Igor A. *Experience in theories of Meaning ⇔ Text linguistic models* (in Russian). Moscow: Nauka, 1974.
14. [Miller, 1990] Miller, George A., ed. *WordNet: An on-line lexical database*. International Journal of Lexicography, 3, 1990: pp. 235-312.
15. [Narin'yani, 1997] Narin'yani, A.S. *Automatic text understanding – new perspective* (in Russian, abstract in English). In Proceedings of International Workshop Dialogue'97: Computational Linguistics and its Applications, Moscow, 1997.
16. [Patwardhan et al. 2003] Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. *Using Measures of Semantic Relatedness for Word Sense Disambiguation*. In Proceedings of CICLing 2003, the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, pp. 241–257, 2003, Mexico City.
17. [Pedersen et al., 2004] Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi, J. 2004. *WordNet::Similarity: Measuring the relatedness of concepts*. In HLT-NAACL 2004, Association for Computational Linguistics, pp. 38–41.
18. [Shier, 1976] Shier D.R. *Iterative methods for determining the* K *shortest paths in a network*. In Networks, 1976. Vol. 6, N 3, pp. 205-229.
19. [Sidorov et al., 2011] Sidorov, Grigori, Juan-Pablo Posadas-Durán, Héctor Jiménez-Salazar, Liliana Chanona-Hernández. *A New Combined Lexical and Statistical based Sentence Level Alignment Algorithm for Parallel Texts*. International Journal of Computational Linguistics and Applications, Vol 2 (1-2), 2011, pp. 257–263.
20. [Small and Rieger, 1982] Small, S.L., and C.J. Rieger. *Parsing and comprehending with word experts (a theory and its realization)*. In Lehnert and Ringle (eds.), Strategies for Natural Language Processing, 1982, pp. 89-147.
21. [Spanish-English, 1963] *Spanish-English, English-Spanish dictionary*, Pocket books, Inc. NY, 1963.
22. [Steel, 1990] Steel, James, ed. *Meaning – Text Theory. Linguistics, lexicography, and implications*. University of Ottawa press, 1990.
23. [Sussna, 1993] Sussna, M. *Word Sense disambiguation for free text indexing using a massive semantic network*. In: Proceedings of CIKM, 1993.
24. [Voorhees, 1993] Voorhees, E.M. *Using WordNet to disambiguate word sense for text retrieval*. In Proceedings of ACM SIGIR Conference, 1993, pp. 171-180.
25. [Yarowsky, 1995] Yarowsky, David. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*. In Proceedings of the 33rd Annual Meeting of the Amer. Soc. for Comp. Ling., 1995, pp. 189-196.
26. [Yarowsky, 1992] Yarowsky, David. Word Sense Disambiguation Using Statistical Models of Roget's Categories Training on Large Corpora. In Proceedings of COLING-92, 1992, pp. 454-460.

Estudio sobre métodos tipo Lesk usados para la desambiguación de sentidos de palabras

Sulema Torres-Ramos

Centro de Investigación en Computación (CIC-IPN),
Unidad Profesional Adolfo López Mateos,
Av. Juan de Dios Bátiz s/n esquina M. Othón de Mendizábal,
Zacatenco, México, D.F. 07738, México.
sulema7@gmail.com

Resumen. La ambigüedad semántica es un problema que se presenta en todos los lenguajes naturales. Podríamos decir que para los seres humanos la ambigüedad en el lenguaje pasa desapercibida, debido a que la resolvemos casi inconscientemente utilizando la realidad en que vivimos, el contexto y el conocimiento que poseemos sobre algunos temas. Pero para las computadoras no es así. En el área de procesamiento de lenguaje natural, la tarea de desambiguación de sentidos de palabras es el problema de seleccionar un sentido, de un conjunto de posibilidades predefinidas, para una palabra dada en un texto o discurso. La desambiguación del sentido de las palabras, es considerada como uno de los problemas más importantes de investigación en el procesamiento del lenguaje natural. Es esencial para las aplicaciones que requieren la comprensión del lenguaje, como la comunicación hombre-máquina, traducción automática, recuperación de la información y otros. Uno de los primeros métodos propuestos para llevar a cabo esta tarea es el método de Lesk, el cual propone utilizar la coherencia global del texto, es decir, el total de sentidos de palabras relacionadas en el texto. La ventaja de este método es que sólo se necesita un diccionario de sentidos como recurso léxico. El problema principal es que mientras más palabras tengamos, más grande es el espacio de búsqueda. Por lo tanto, se han desarrollado diferentes métodos (conocidos como métodos tipo Lesk) que aplican modificaciones de este algoritmo intentando obtener la combinación de sentidos óptima para un texto dado. En este artículo se presenta un estudio de los principales métodos tipo Lesk usados para la desambiguación de sentidos de palabras.

Palabras clave: Lingüística Computacional, Procesamiento de Lenguaje Natural, Desambiguación de Sentidos de Palabras, Algoritmo de Lesk.

1 Introducción

La información es el recurso más importante que poseemos los seres humanos. Gran parte de esta información se comunica, almacena y maneja en forma de lenguaje natural (español, inglés, ruso, etc.). En la actualidad, podemos obtener grandes volúmenes de información en forma escrita, ya sea de manera impresa o electrónica.

Las computadoras son una herramienta indispensable para el procesamiento de la información plasmada en los textos, ya que son capaces de manejar grandes volúmenes de datos. Sin embargo, una computadora no puede hacer todo lo que las personas normalmente hacemos con el texto, por ejemplo, responder preguntas basándose en la información proporcionada, o, hacer inferencias lógicas sobre su contenido, o elaborar un resumen de esta información.

Por lo anterior, el Procesamiento de Lenguaje Natural (PLN) tiene por objetivo habilitar a las computadoras para que entiendan el texto, procesándolo por su sentido.

Para llevar a cabo esta tarea, un sistema de PLN necesita conocer sobre la estructura del lenguaje, la cual se analiza normalmente en los siguientes niveles [1]:

- Nivel fonológico: trata de los sonidos que componen el habla, permitiendo formar y distinguir palabras.
- Nivel morfológico: trata sobre la estructura de las palabras y las leyes para formar nuevas palabras a partir de unidades de significado más pequeñas llamadas morfemas.
- Nivel sintáctico: trata sobre cómo las palabras pueden unirse para construir oraciones y cuál es la función que cada palabra realiza en esa oración.
- Nivel semántico: trata del significado de las palabras y de cómo se unen para dar significado a una oración.
- Nivel pragmático: estudia la intención del hablante al producir oraciones específicas o textos en una situación específica.

Todos los niveles anteriores de la estructura del lenguaje tienen un problema: la ambigüedad.

La ambigüedad, en el proceso lingüístico, se presenta cuando pueden admitirse distintas interpretaciones a partir de una representación dada o cuando existe confusión al tener diversas estructuras y no tener los elementos necesarios para eliminar las eventualmente incorrectas. Para desambiguar, es decir, para seleccionar los significados o las estructuras más adecuadas de un conjunto conocido de posibilidades, se requieren diversas estrategias de solución según el caso [2].

Debido a que existe ambigüedad aún para los humanos, su solución no es sólo lograr la asignación del sentido único por palabra en el análisis de textos, sino eliminar la gran cantidad de variantes que normalmente existen. La ambigüedad es el problema más importante en el procesamiento de textos en lenguaje natural, por lo que su resolución es la tarea más importante a llevar a cabo.

Se distinguen tres tipos principales de ambigüedad: léxica, sintáctica (estructural) y semántica.

La ambigüedad léxica se presenta cuando las palabras pueden pertenecer a diferentes categorías gramaticales, por ejemplo, la palabra *bajo* puede ser una preposición, un sustantivo, un adjetivo o una conjugación del verbo bajar [3].

La ambigüedad sintáctica, también conocida como ambigüedad estructural se presenta cuando una oración puede tener más de una estructura sintáctica. Por ejemplo de la oración "*María habló con el profesor del instituto*" se puede entender dos cosas diferentes: a) el profesor pertenece al instituto, o bien, b) el tema del que habló María con el profesor fue el instituto [4].

La ambigüedad semántica se presenta cuando las palabras tienen múltiples significados, por ejemplo la palabra *banco* puede significar institución financiera, la orilla del lago, asiento, etc.

Hoy en día, cualquier palabra que usamos para comunicarnos tiene dos o más posibles interpretaciones, llamadas sentidos. Para entender correctamente un texto, el lector –humano o programa de computadora– debe ser capaz de determinar el sentido adecuado para cada palabra en el texto.

Además de entender un texto, hay muchas aplicaciones de procesamiento de lenguaje natural donde la determinación automática del sentido correcto de una palabra es crucial. Entre ellas se encuentran:

1. Traducción automática. La desambiguación semántica es esencial para la traducción apropiada de palabras como *bank*(banco) que, dependiendo del contexto, puede traducirse como *institución bancaria*, *orilla del río*, etc. [5,6].
2. Recuperación de información. Al realizar búsquedas por palabras clave específicas, es necesario eliminar los documentos donde se usa la palabra o palabras en un sentido diferente al deseado; por ejemplo, al buscar referencias sobre animales con la palabra *gato*, es deseable eliminar los documentos que contienen dicha palabra asociada con mecánica automotriz [7,8,9,10,11,20,21].
3. El procesamiento de texto. La desambiguación es necesaria para algunas tareas de procesamiento de texto, por ejemplo, para determinar cuándo deben insertarse acentos diacríticos [12,13] y para la detección y corrección del malapropismo [14,15,16].
4. Respuesta automática a preguntas (QA, por sus siglas en inglés: Question Answering): La meta de esta tarea es encontrar respuestas en el dominio de texto en lenguaje natural [17,18]. A diferencia de un sistema de recuperación de información que te devuelve los documentos relativos a un criterio de búsqueda, un sistema de QA devolverá una respuesta específica a la búsqueda especificada. Ejemplo: si buscamos "Patente bulbo Edison" un sistema de recuperación de información devolverá la lista de documentos relevantes que tengan que ver con la patente de bulbos de Edison, sin embargo un sistema de QA preguntará "¿Cuándo se registró la patente de bulbo de Edison?" y el sistema devolverá una respuesta específica.

En el área de procesamiento de lenguaje natural o lingüística computacional, la identificación del sentido de palabras en un contexto dado es conocida como desambiguación de sentidos de palabras (WSD por sus siglas en inglés).

Una forma de llevar a cabo la desambiguación de sentidos de palabras es tomar en cuenta la coherencia global del texto [19], es decir, el total de sentidos de palabras relacionadas en el texto. La limitación principal de esta técnica es que, para encontrar la combinación óptima de sentidos se necesita mucho tiempo, ya que el espacio de búsqueda es muy grande. La principal ventaja de esta técnica es que es un método no supervisado y sólo utiliza un diccionario de sentidos como recurso externo.

Debido a lo anterior, existen diferentes métodos (conocidos como métodos tipo Lesk) que aplican modificaciones del algoritmo original de Lesk para desambiguar palabras en un texto.

En este artículo se presenta una revisión de los principales métodos tipo Lesk usados para llevar a cabo la desambiguación de sentidos de palabras como una tarea del procesamiento de lenguaje natural.

El artículo se organiza como sigue: primero, formalizamos la tarea de desambiguación de sentidos de palabras y los métodos usados para esta tarea clasificados de acuerdo a los recursos que utilizan (sección 2). Después, se explica a detalle el algoritmo original de Lesk (sección 3) y se presenta un estudio sobre los principales métodos que se basan en dicho algoritmo (sección 4). En la sección 5 se presenta un análisis de los resultados obtenidos para los métodos tipo Lesk y al final se presentan las conclusiones.

2 Desambiguación del sentido de las palabras (WSD)

En general, la desambiguación del sentido de las palabras es el problema de seleccionar un sentido de un conjunto de posibilidades predefinidas para una palabra dada en un texto o discurso.

En los últimos años se han incrementado las investigaciones para crear métodos de WSD. A continuación se describe la clasificación para métodos de WSD de acuerdo a los recursos que utilizan (figura 1).

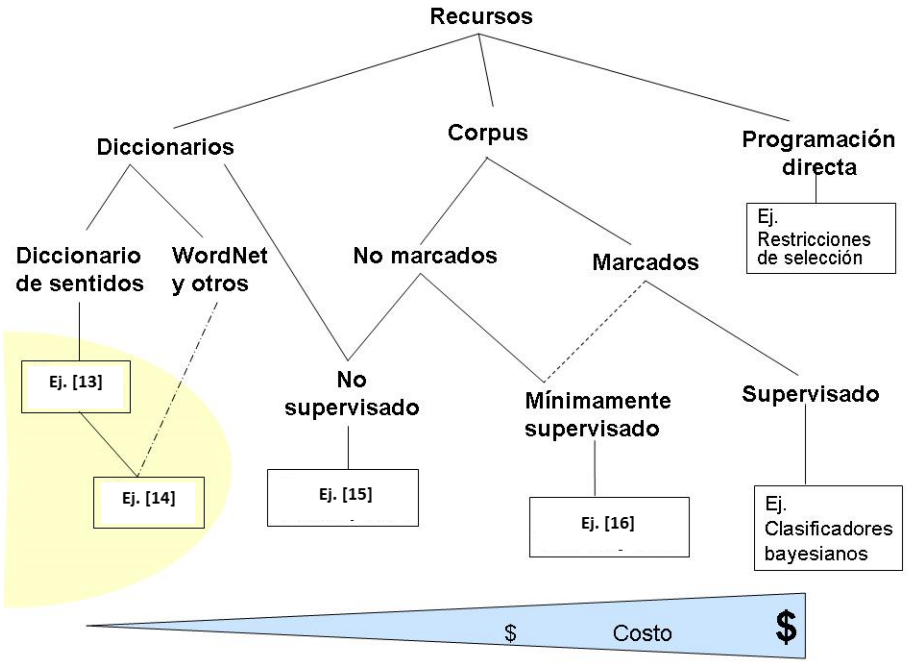

Fig. 1. Clasificación de los métodos para WSD de acuerdo a los recursos que utilizan.

2.1 Clasificación de métodos para desambiguación de sentidos de palabras

Los métodos para desambiguación de sentidos de palabras se clasifican en: los que utilizan diccionarios, los que utilizan corpus, y los que no utilizan ningún recurso léxico.

Los que utilizan ***diccionarios***:

Los diccionarios pueden ser *de sentidos* y otros como *WordNet*.

Los diccionarios proporcionan una lista de glosas (definición de sentido) para las palabras. Los métodos que utilizan sólo diccionarios de sentidos, buscan elegir un sentido (de esta lista) para cada palabra en un texto dado, tomando en cuenta el contexto en el que aparece. Como ejemplo, [19] propone utilizar la coherencia global del texto, es decir, el total de sentidos de palabras relacionadas en el texto: mientras más relacionadas estén las palabras entre sí, más coherente será el texto.

Además existen variantes del algoritmo de Lesk que utilizan no sólo diccionarios de sentidos, sino también otro tipo de diccionarios como WordNet.

Los que utilizan ***corpus***:

Los corpus pueden ser *no marcados* y *marcados*.

Los métodos que utilizan corpus no marcados son los no supervisados, estos métodos también utilizan otros recursos como WordNet para poder asignar un sentido a cada palabra que aparece en los textos no marcados. Como ejemplo de éstos tenemos el método de [22], el cual elige de un diccionario (tesauro) las palabras relacionadas con la palabra a desambiguar. Cada palabra relacionada tiene un peso, éstas y la palabra a desambiguar tienen sentidos en un diccionario. Para elegir el sentido correcto, las palabras relacionadas votan por un sentido de la palabra a desambiguar con cierto peso. Se elige el sentido con más peso.

Los métodos que utilizan corpus marcados son los métodos supervisados. Éstos reducen la desambiguación de sentidos de palabras a un problema de clasificación, donde a una palabra dada se le asigna el sentido más apropiado de acuerdo a un conjunto de posibilidades, basadas en el contexto en el que ocurre. Hay muchos algoritmos de aprendizaje supervisado utilizados para WSD, como ejemplo tenemos los clasificadores bayesianos, máquinas de soporte vectorial, árboles y listas de decisión, etc. [23].

Hay métodos que utilizan una gran cantidad de corpus no marcados [24] y muy pocos marcados [25] llamados mínimamente supervisados. Como ejemplo de éstos tenemos el método de [26], el cual identifica todas las ocurrencias de una palabra a desambiguar en un corpus no marcado. Después identifica un número pequeño de colocaciones semilla representativos de cada sentido de la palabra y etiqueta todos los ejemplos que contienen la colocación semilla con la palabra de dicha colocación (así tenemos los conjuntos etiquetados con cada sentido representativo y el conjunto residuo). El algoritmo utiliza los conjuntos etiquetados para entrenar una lista de decisión y encontrar nuevas colocaciones, para después etiquetar sobre el conjunto residuo. El algoritmo termina cuando el conjunto residuo se estabiliza.

Los que utilizan ***programación directa***:

Estos métodos se basan en reglas (muchas) que especifican el sentido de una palabra de acuerdo al contexto en el que aparece. Un ejemplo son las restricciones de selección (selectional restrictions), definen reglas de acuerdo a la palabra a desambiguar y su argumento. Ejemplo: el verbo *comer* puede tener como restricción que su tema argumento sea comida (comer-comida).

Este artículo se enfoca en los métodos que se basan en la ***aplicación directa de diccionarios de sentidos***, que se describen a continuación

3 Algoritmo de Lesk

El algoritmo de Lesk [19] es uno de los primeros algoritmos exitosos usados en la desambiguación de sentidos de palabras. Este algoritmo se basa en dos puntos principales: un algoritmo de optimización para WSD y una medida de similitud para las definiciones de los sentidos.

El primer punto es acerca de desambiguar palabras considerando la coherencia global del texto, esto es, encontrar la combinación de los sentidos que maximice la relación total entre los sentidos de todas las palabras.

Por ejemplo, para la oración *My father deposits his money in a bank account* y considerando a lo más tres sentidos[1] (véase tabla 1), para cada palabra, la figura 2 muestra la representación gráfica del algoritmo original de Lesk.

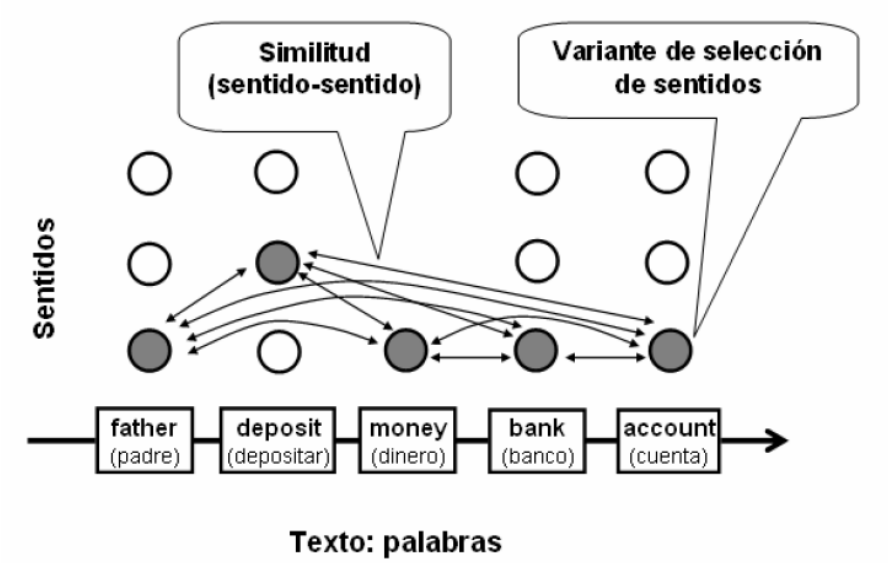

Fig. 2. Representación gráfica del algoritmo original de Lesk.

[1] Sentidos obtenidos de WordNet

Tabla 1. Sentidos de las palabras (máximo tres) obtenidas de WordNet para la oración "*My father deposits his money in a bank account*".

Palabra	Sentidos
Father	1: a male parent (also used as a term of address to your father); "his father was born in Atlanta". 2: `Father' is a term of address for priests in some churches (especially the Roman Catholic Church or the Orthodox Catholic Church); "`Padre' is frequently used in the military". 3: a person who holds an important or distinguished position in some organization; "the tennis fathers ruled in her favor"; "the city fathers endorsed the proposal".
Deposit	1: fix, force, or implant; "lodge a bullet in the table". 2: put into a bank account; "She deposits her paycheck every month". 3: put (something somewhere) firmly; "She posited her hand on his shoulder"; "deposit the suitcase on the bench"; "fix your eyes on this spot".
Money	1: the official currency issued by a government or national bank; "he changed his money into francs".
Bank	1: a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home". 2: sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents". 3: a supply or stock held in reserve for future use (especially in emergencies)
Account	1: a formal contractual relationship established to provide for regular banking or brokerage or business services; "he asked to see the executive who handled his account". 2: the act of informing by verbal report; "he heard reports that they were causing trouble"; "by all accounts they were a happy couple". 3: a record or narrative description of past events; "a history of France"; "he gave an inaccurate account of the plot to kill the president"; "the story of exposure to lead".

En el segundo punto, relacionado con la medida de similitud, Lesk sugiere usar el *traslape* entre las definiciones de los sentidos, es decir, contar el número de palabras que tienen en común.

Como ejemplo, para la oración, "*My father deposits his money in the bank account*" para medir la relación de las definiciones de los sentidos para la palabra "*deposit*" y "*bank*" como Lesk lo propuso, es necesario contar las palabras en común en todas las definiciones. En este caso, comparando principalmente las tres definiciones de "*deposit*" contra las tres definiciones de "*bank*". La relación entre los valores se muestra en la tabla 2.

Tabla 2. Valores de relación para las definiciones de sentidos de las palabras "*deposit*" y "*bank*".

Sentido elegido para *deposit*	Sentido elegido para *bank*	Valor de relación (traslape de palabras)
1	1	0
1	2	0
1	3	0
2	**1**	**2**
2	2	1
2	3	0
3	1	1
3	2	0
3	3	0

Este algoritmo tiene dos limitaciones, por un lado, la limitación principal de la medida de similitud propuesta por Lesk, es que las glosas del diccionario, regularmente, son muy cortas y no incluyen el vocabulario suficiente para identificar los sentidos relacionados [27].

Por otro lado, mientras más palabras tenga el texto, y más sentidos por cada palabra, mayor será el número de combinaciones de sentidos, haciéndolo prácticamente prohibitivo para una búsqueda exhaustiva que garantice encontrar el óptimo global exacto. Por ejemplo, para una oración de 16 palabras de contenido, donde cada palabra contiene siete sentidos (números cercanos a los observados en el corpus de SemCor), existen 7^{16} posibles combinaciones a escoger, de las cuales una será seleccionada.

Debido a estas dos limitaciones, diferentes modificaciones al algoritmo original han sido propuestas para mejorar los resultados en la desambiguación de sentidos de palabras, las cuales se describen en la siguiente sección.

4 Métodos tipo Lesk

Las modificaciones al algoritmo original de Lesk han sido propuestas tanto del lado de la medida de similitud como del lado del problema de complejidad computacional.

4.1 Basados en medidas de similitud semántica

Como se mencionó anteriormente la medida de similitud propuesta por Lesk, traslape (overlapping), tiene la limitación del tamaño de las glosas del diccionario que, regularmente son muy cortas.

A continuación se describen ciertas medidas de similitud las cuales han sido propuestas para medir la proximidad semántica entre dos sentidos o palabras, usando WordNet como espacio semántico.

Medida de Lesk Adaptada

Lesk propuso medir la similitud entre sentidos contando el traslape de palabras. La limitación principal de esta técnica es que las glosas del diccionario, por lo general, son muy breves, de tal manera que no incluyen suficiente vocabulario para identificar los sentidos relacionados. En [28] se sugiere una adaptación del algoritmo basado en WordNet. Esta adaptación consiste en tomar en cuenta las glosas de los vecinos de la palabra a desambiguar, explotando los conceptos jerárquicos de WordNet, de tal manera que las glosas de los vecinos son expandidas incluyendo a su vez las glosas de las palabras con las cuales se encuentran relacionadas mediante las diversas jerarquías que presenta WordNet. Así mismo, sugieren una variación en la manera de asignar el puntaje a una glosa, de tal manera que si "n" palabras consecutivas son iguales en ambas glosas, estas deberán de tener mayor puntaje que aquel caso en el que sólo coincide una sola palabra en ambas glosas.

Supongamos que *bark* (ladrido o corteza) es la palabra que se desea desambiguar y sus vecinos son *dog* (perro) y *tail* (cola). El algoritmo original de Lesk verifica las coincidencias en las glosas de los sentidos de *dog* con las glosas de *bark*. Luego verifica las coincidencias en las glosas de *bark* y *tail*. El sentido de *bark* con el máximo número de coincidencias es seleccionado. La adaptación del algoritmo de Lesk considera estas mismas coincidencias y añade además las glosas de los sentidos de los conceptos que se encuentran relacionados semántica o léxicamente a *dog*, *bark* y *tail*, de acuerdo a las jerarquías de WordNet.

Medida de Leacock-Chodorow

Esta medida está basada en las longitudes de rutas usando la jerarquía "es-un" de WordNet, para las definiciones de sustantivos [29]. La ruta más corta entre dos conceptos es aquella que incluye el menor número de conceptos intermedios.

Este valor es escalado por la profundidad de la jerarquía, donde dicha profundidad es definida como la longitud desde el nodo raíz hasta un nodo hoja. Por consiguiente la medida de relación está definida por la siguiente fórmula:

$$Similitud_{lch}(C_1, C_2) = \max[- \log(RutaMasCorta(C_1, C_2) /(2.D))] \quad (1)$$

$RutaMasCorta(C_1, C_2)$ es la longitud de la ruta más corta entre dos conceptos (ruta con menor número de nodos) y D es la profundidad máxima de la taxonomía (distancia entre la raíz y el nodo más alejado de ésta). La implementación de esta medida usando WordNet, asume un nodo raíz hipotético que junta todas las jerarquías de sustantivos, de tal manera que D llega a ser una constante de 16 para todos los sustantivos, lo cual significa que la longitud entre el nodo raíz y la hoja más lejana del árbol es de 16.

Medida de Resnik

Resnik [30] introduce una medida de relación basada en el concepto de "contenido de la información" más conocido en inglés como information content, el cual se trasluce

como un valor que es asignado a cada concepto en una jerarquía basada en la evidencia encontrada en un corpus.

El término "contenido de la información" es una simple medida de la especificación de un concepto. Un concepto con un gran contenido de información es muy específico a un tópico particular, mientras que conceptos con un contenido de información bajo están asociados con tópicos más generales. Por lo tanto, la expresión *carving fork* (tenedor) tiene un alto contenido de información, mientras que *entity* (entidad) tiene un bajo contenido de información.

El contenido de información de un contexto es estimado contando la frecuencia de ese concepto en un corpus de gran escala, determinando de esta manera su probabilidad. De acuerdo a Resnik, el logaritmo negativo de esta probabilidad determina el contenido de información del concepto.

$$IC(concept) = - \log(P(concept)) \quad (2)$$

Si se tuviera un texto etiquetado de sentidos, contar la frecuencia de un concepto sería logrado directamente, ya que cada concepto sería asociado con un único sentido; pero en caso contrario, Resnik sugiere contar el número de ocurrencias de una palabra en el corpus y luego dividir dicho valor por el número de sentidos que tiene dicho término, siendo este valor asignado a cada concepto. Por ejemplo, supongamos que la palabra *bank* (banco) ocurre 20 veces en un corpus, y existen dos conceptos asociados a dicha palabra en una jerarquía, uno para *river bank* (orilla de río) y el otro para *financial bank* (institución financiera). Cada uno de estos conceptos recibirá un valor de 10; en cambio si las ocurrencias de *bank*, se presentaran en un texto etiquetado con sentidos, la información sería más consistente.

La frecuencia de un concepto incluye la frecuencia de todos sus conceptos subordinados, ya que el conteo de un concepto es añadido a su inmediato superior. Es necesario notar que los conteos de los conceptos más específicos son añadidos a los más genéricos; y no de manera contraria; por ende los conteos de los conceptos específicos incrementan el total de los más genéricos. Dichos conceptos tendrán una mayor probabilidad asociada, lo que significaría que tendrían un bajo "contenido de información", ya que estos representan conceptos muy generales. La medida de Resnik usa el "contenido de información" de conceptos dentro de las jerarquías "es-un". La idea principal detrás de esta medida es que dos conceptos están semánticamente relacionados teniendo en cuenta la cantidad de información que ellos comparten en común. La cantidad de información común de dos conceptos es determinada por el "contenido de información" del concepto más bajo (*lowest common subsumer*) para las dos conceptos en cuestión. La medida de Resnik es calculada con la siguiente fórmula:

$$Similitud_{res}(C_1, C_2) = IC(lowest\_common\_subsumer(C_1, C_2)) \quad (3)$$

Esta medida no considera el contenido de información del par de conceptos a comparar y tampoco considera la longitud de la ruta entre ambos. La principal limitante de esta técnica es que algunos pares de conceptos compartirían el mismo valor de similitud, ya que existe la posibilidad de que el mismo *lowest common subsumer* sea asignado a más de un par de conceptos. Por ejemplo, *vehicle* (vehículo) es el *lowest com-*

mon subsumer de *jumbo jet* (avión jumbo), *tank* (tanque) y *house trailer* (remolque). Por consiguiente estas parejas recibirían el mismo puntaje en su comparación.

Medida de Jiang-Conrath

Jiang y Conrath [31] usan el concepto de "contenido de información" planteado por Resnik, al cual lo complementan con las longitudes de rutas entre conceptos. Esto resulta una técnica híbrida para computar la relación semántica de una pareja de conceptos. Esta técnica incluye el "contenido de información" de los propios conceptos y del *lowest common subsumer*. Esta medida es determinada por la siguiente fórmula:

$$Similitud_{jcn}(C_1, C_2) = IC(C_1) + IC(C_2) - 2 \times IC(lowest\_common\_subsumer(C_1, C_2)) \quad (4)$$

Medida de Lin

La medida de Lin [32] está basada en su teorema de similitud. Este establece que la similitud de dos conceptos es medida por la razón entre la cantidad de información necesaria para establecer la información común de ambos conceptos y la cantidad de información necesaria para describirlos. Esta información común entre dos conceptos es obtenida por el contenido de información del lowest common subsumer que aplica para ambos conceptos y el contenido de información de cada uno los conceptos propiamente dichos.

Esta medida es muy parecida a la presentada por Jiang y Conrath; aunque ellas fueros desarrolladas independientemente. Esta medida es determinada por la siguiente fórmula:

$$Similitud_{lin}(C_1, C_2) = 2 \times IC(lowest\_common\_subsumer(C_1, C_2)) / IC(C_1) + IC(C_2) \quad (5)$$

Esta medida puede ser vista como la intersección del contenido de información de los dos conceptos a comparar dividido por la suma del contenido de información de ambos.

Medida *vector*

Esta medida, al igual que la de Lesk, incorpora información de las glosas de WordNet. La medida vector, crea una matriz de co-ocurrencia para cada palabra usada en las glosas de WordNet tomando cualquier corpus y luego representa cada glosa con un vector que es el promedio de los vectores de co-ocurrencia.

Medida *path*

Este medida calcula la relación semántica de sentidos contando el número de nodos junto con el camino más corto entre el sentido en la jerarquía "es-un" de WordNet. La longitud de los caminos incluye los nodos terminales.

Dado que un camino largo indica menos relación, el valor de relación obtenido es el inverso multiplicativo de la longitud del camino (distancia) entre los dos conceptos:

relación = 1/distancia. Si los dos conceptos son idénticos, entonces la distancia entre ellos es uno; por lo tanto, su relación también es 1. Sí no es encontrado ningún camino, entonces un valor negativo muy grande es devuelto.

Medida combinada

En [33] proponen implementar una combinación de medidas de similitud dependiendo del par de categorías gramaticales a ser medidos. Ellos recomiendan utilizar la medida JCN para obtener la similitude entre sustantivos y la medida LCH para verbos. Para todas las demás relaciones se recomienda usar la medida de LESK.

4.2 Basados en la complejidad computacional

Para aliviar el problema de la complejidad computacional del algoritmo original de Lesk, dos principales soluciones han sido propuestas, a) una versión simplificada que considere las palabras, una por una, y compare cada sentido de la palabra dada con el contexto, y b) el uso de búsquedas basadas en heurísticas para encontrar una solución óptima cercana a la real en un menor tiempo [34].

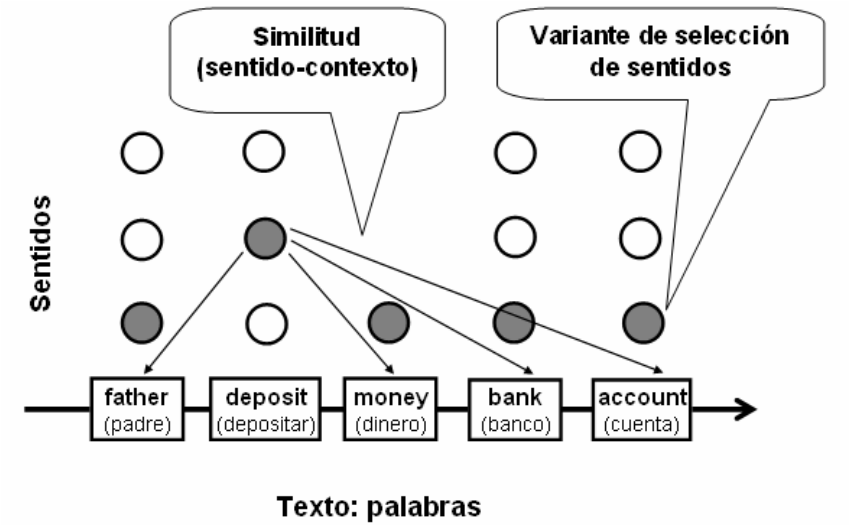

Fig. 3. Representación gráfica del algoritmo de Lesk simplificado.

Lesk simple o Lesk simplificado

Para reducir el espacio de búsqueda del algoritmo original de Lesk, Kilgarriff y Rosenzweig [35] propusieron una variación del algoritmo original de Lesk, conocido como algoritmo de **Lesk simplificado** o **Lesk simple**, donde los sentidos de las palabras en el texto son determinados uno a uno encontrando el mayor traslape entre los sentidos de las definiciones de cada palabra con el contexto actual, véase la figura 3.

En lugar de buscar asignar, simultáneamente, el significado de todas la palabras en un texto dado, este enfoque determina el sentido de las palabras uno a uno, por lo que se evita la explosión combinatoria de sentidos.

Templado simulado (Simulated Annealing)

El método de templado simulado es una técnica para la resolución de problemas de optimización combinatoria a gran escala. El nombre de este algoritmo es una analogía del proceso metalúrgico en el cuál, el metal se enfría y se templa. La característica de este fenómeno es que en el enfriamiento lento alcanza una composición uniforme y un estado de energía mínimo, sin embargo, cuando el proceso de enfriamiento es rápido, el metal alcanza un estado amorfo y con un estado alto de energía. En templado simulado la variable **T** corresponde a la temperatura que decrece lentamente hasta encontrar el estado mínimo.

El proceso requiere una función **E**, la cual representa el estado de energía de cada configuración del sistema. Es esta función la que se intenta minimizar. A grandes rasgos el algoritmo funciona de la siguiente manera: se selecciona un punto inicial y además se escoge otra configuración de manera aleatoria, se calcula para ambas configuraciones su valor **E**, si el nuevo valor es menor que el seleccionado como punto inicial, entonces el inicial es remplazado por la nueva configuración. Una característica esencial del templado simulado es que, existe el caso en el que la nueva configuración es mayor a la configuración obtenida anteriormente, y la nueva es seleccionada. Esta decisión es tomada de manera probabilística y permite salir de algún mínimo local. Una vez que el método mantenga la misma configuración por un determinado tiempo, dicha configuración es escogida como la solución.

Cowie et al. [36], basándose en el algoritmo de Lesk, utilizó este método para desambiguación de sentidos de palabras de la siguiente forma:

1. El algoritmo define una función **E** para la combinación de sentidos de palabras en un texto dado.
2. Se calcula **E** para la configuración inicial **C**, donde **C** es el sentido más frecuente para cada palabra.
3. Para cada iteración, se escoge aleatoriamente otra configuración conocida como **C'**, y se calcula su valor de **E**. Si el valor de **E** para **C'** es menor que el de **C** entonces se elige **C'** como configuración inicial.
4. La rutina termina cuando la configuración de sentidos no ha cambiado en un tiempo determinado.

Algoritmos genéticos

Introducidos por Holland [37] e impulsados en años sucesivos por Goldberg [38], uno de sus estudiantes, los algoritmos genéticos han sido utilizados con éxito en múltiples campos de la ciencia. Los algoritmos genéticos son métodos sistemáticos para la resolución de problemas de *búsqueda y optimización*, que aplican a éstos los mismos mé-

todos de la evolución biológica: selección basada en la población, reproducción sexual y mutación.

En un algoritmo genético, tras parametrizar el problema en una serie de variables, (x_i,...,x_n) se codifican en un cromosoma. Todos los operadores utilizados por un algoritmo genético se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos. En el algoritmo genético va implícito el método para resolver el problema; son solo parámetros de tal método los que están codificados, a diferencia de otros algoritmos evolutivos como la programación genética. Hay que tener en cuenta que un algoritmo genético es independiente del problema, lo cual lo hace un algoritmo *robusto*, por ser útil para cualquier problema, pero a la vez *débil*, pues no está especializado en ninguno.

Las soluciones codificadas en un cromosoma *compiten* para ver cuál constituye la mejor solución (aunque no necesariamente la mejor de todas las soluciones posibles). El *ambiente*, constituido por otras soluciones, ejercerá una presión selectiva sobre la población, de forma que sólo los mejor adaptados (aquellos que resuelvan mejor el problema) sobrevivan o leguen su material genético a las siguientes generaciones, igual que en la evolución de las especies. La diversidad genética se introduce mediante mutaciones y reproducción sexual.

En la naturaleza lo único que hay que optimizar es la supervivencia, y eso significa a su vez maximizar diversos factores y minimizar otros. Un algoritmo genético, sin embargo, se usará habitualmente para optimizar sólo una función, no diversas funciones relacionadas entre sí simultáneamente. La optimización que busca diferentes objetivos simultáneamente, denominada multimodal o multiobjetivo, también se suele abordar con un algoritmo genético especializado.

Por lo tanto, un algoritmo genético consiste en lo siguiente: hallar de qué parámetros depende el problema, codificarlos en un cromosoma, y se aplican los métodos de la evolución: selección y reproducción sexual con intercambio de información y alteraciones que generan diversidad.

En [39] utilizaron un algoritmo genético que elige los sentidos que dan más coherencia al texto en términos de medidas de relación de palabras. El método optimiza globalmente el total de relaciones de palabras y no cada palabra de manera independiente [40].

Los parametros del algoritmo que utilizaron fueron los siguientes:

- El cromosoma es una secuencia de números naturales de 1 a n_i, donde n_i es el número de sentidos de la palabra w_i. Si la palabra no tiene sentidos, es decir, no se encuentra en el diccionario, la posición correspondiente en el cromosoma no se usa.
- El contenido inicial de la *piscina* fue generado aleatoriamente: para cada individuo y cada posición i en su cromosoma, el valor fue generado aleatoriamente con distribución uniforme en el dominio entre 1 y n_i.
- La función de aptitud (*fitness function*) fue definida por la siguiente fórmula:

```
for each sequence f ∈ F
  for each word w_k
      score(w_k) = Σ_i M_{i,f(w_i)}(w, f(k))
  score(f) = Σ_k^N score(w_k)
f_best = max arg (score(f))
for each word w_k
  select s_best = f_best(k)
```

- Se utilizaron dos *piscinas*, de modo que en cada generación todos los padres fueron sustituidos por sus respectivos descendientes, por lo que no hay individuos de una nueva generación que se apareen con individuos de las anteriores. Esto también significa que el método de sustitución se fijó a anexar: los nuevos individuos se anexan a la nueva *piscina*.
- No se utilizó ninguna brecha generacional, es decir, hay un número predefinido de individuos clonados en la nueva generación
- El método de selección fue *roulette wheel*: la probabilidad de seleccionar un individuo para el cruce (o clonación) es proporcional a su valor de aptitud
- El esquema de generación fue el siguiente: el par seleccionado de los padres fue reemplazado con dos descendientes formados por el intercambio de las partes seleccionadas de los cromosomas de los padres. Con cierta probabilidad de que los padres sean clonados en la nueva generación en lugar de ser acoplados.
- La probabilidad de cruce fue determinada por el parámetro llamado *crossover rate* (tasa de cruce) que controlaba la opción de cruce: si dos individuos seleccionados se cruzaban, dos descendientes se formaban como resultado o simplemente los dos padres se clonaban en la nueva generación. Entre más grande la tasa de cruce, mayor es la probabilidad de que los padres se apareen.
- El método de cruce es simple: un solo punto de cruce es seleccionado aleatoriamente; los genes incluidos hasta el punto de cruce se copian en el hijo respectivo y el resto de los genes fueron copiados a un hijo alternativo.
- El esquema de mutación es como sigue: cada hijo fue seleccionado o no para mutación con la probabilidad determinada por el parámetro llamado tasa de mutación (*mutation rate*). Si se selecciona, un único punto de mutación *i* fue seleccionado al azar(con distribución uniforme)
- Una mutación en un punto *i* fue un cambio al azar de un gen en su respectivo dominio, es decir, de 1 a n_i, donde n_i es el número de sentidos de la palabra w_i.
- Se usó elitismo para acelerar la convergencia. Esto implica las dos modificaciones siguientes en el comportamiento estándar del algoritmo. En primer lugar, dos copias del mejor individuo se clonan a la *piscina* de la nueva generación, asegurando así su supervivencia. En segundo lugar, en cada acción de cruce, de cada cuatro individuos -los dos padres y dos hijos- dos mejores se colocan en la nueva *piscina*. De esta manera, si un hijo no es tan bueno como cualquiera de los padres, no va a ser seleccionado, y uno de los padres va a sobrevivir en su lugar.
- La condición de término es la convergencia: el algoritmo se detiene cuando todos los individuos en el grupo tienen el mismo valor de aptitud (*fitness*).

5 Análisis de resultados

En esta sección se describen las evaluaciones llevadas a cabo para la desambiguación de sentidos de palabras con métodos tipo Lesk (reportados en el estado del arte) y sus principales resultados.

Para el algoritmo original de Lesk:

Lesk evaluó su algoritmo sobre ejemplos cortos extraídos de "*Pride and Prejudice"* y *"An Associated Press news story*" usando el diccionario "*Oxford Advanced Learner's Dictionary*", con una precisión de 50 a 70 %.

El trabajo de [41] presenta una evaluación del algoritmo original de Lesk con *back-off*[2] a sentido más frecuente sobre SENSEVAL-2 (*English-all words*) y SemCor (20964 instancias), utilizando *WordNet* como diccionario y con una *ventana de contexto*[3] de 2,3,8,10 y 25 palabras. Los resultados fueron similares en ambos corpus con un 43% de precisión.

Para los métodos tipo Lesk que sólo varían la medida de similitud semántica:

La medida adaptada de Lesk [28] fue evaluada sobre los datos de Senseval-2 (*English lexical sample tas*k) y utilizando el diccionario *WordNet*. En esta evaluación se utilizó una *ventana de contexto* de dos palabras y los resultados se reportan de acuerdo a la categoría gramatical de la palabra desambiguada. Para los sustantivos se reporta una precisión del 32.2%, para los verbos 24.9% y para los adjetivos 46.9%; con una precisión general de 31.7%.

En [42] evaluaron tres medidas de similitud (Jiang-Conrath, Lesk y combinada), utilizando el diccionario *WordNet*, sobre los datos de Senseval-2, Senseval-3, Semeval (*Eglish all-words data sets*) y el corpus Semcor, eliminado las oraciones con más de 210,567,168,000 combinaciones. Se utilizó una *ventana de contexto* del tamaño de la oración y evaluaron con dos *métodos de back-off*, sentido más frecuente[4] y sentido aleatorio. Los resultados experimentales mostraron que la medida combinada es más precisa que cada medida por separado. También mostraron que la medida de Lesk tiene mejor desempeño cuando el *back-off* es sentido aleatorio, mientras que la medida de Jiang-Conrath muestra mejores resultados sólo cuando se usa *back-off* a sentido más frecuente.

Para el algoritmo de Lesk simplificado:

En la propuesta original de Lesk Simple, Kilgarrif y Rosenzweig evaluaron el algoritmo sobre los datos de SENSEVAL con los sentidos obtenidos del diccionario léxi-

---

[2] Cuando método principal no tiene suficiente información para elegir el sentido de una palabra, el método de back-off toma la decisión. Ej. cuando el método de back-off es sentido aleatorio, si el algoritmo de Lesk no pudo elegir un sentido entonces se elige cualquier sentido de los posibles

[3] Número de palabras que se encuentran en el texto, antes y después de la palabra a ser desambiguada, y que se tomarán en cuenta en el proceso de desambiguación

[4] De acuerdo con WordNet.

co HECTOR. Se llevó a cabo la comparación entre sentidos utilizando por un lado la glosa del diccionario y por otro, la glosa y los ejemplos. El algoritmo presentó mejores resultados cuando se utiliza la glosa y los ejemplos, con un 55% de precisión, mientras que utilizando sólo la glosa se obtuvo un 30% de precisión.

Así mismo, en [41] se presenta también la evaluación del algoritmo de Lesk simplificado con *back-off* a sentido más frecuente sobre SENSEVAL-2 (English-all words) y SemCor (20964 instancias), utilizando WordNet como diccionario y con una ventana de contexto de 2,3,8,10 y 25 palabras. Los resultados fueron similares en ambos corpus con un 55% de precisión.

En [43] se evalúo el algoritmo de Lesk simplificado con *back-off* a sentido aleatorio sobre tres diferentes corpus usando *WordNet* como diccionario: SENSEVAL-2, SENSEVAL-3 y una muestra de 10 oraciones seleccionadas aleatoriamente del corpus Semcor. La ventana de contexto usada para los experimentos fue el tamaño de la oración y la medida de similitud usada fue el traslape pero fue normalizado, dividiendo entre largo de las definiciones, para evitar la ventaja de definiciones largas. Los resultados obtenidos fueron similares para los tres corpus con aproximadamente 48% de precisión.

Para los métodos tipo Lesk basados en heurísticas para encontrar una solución óptima en menor tiempo:

En [36] se evaluó el algoritmo de Lesk, usando templado simulado, sobre 50 oraciones etiquetadas manualmente y utilizando el diccionario LDOCE (Longman Dictionary of Contemporary English). Las oraciones tenían de dos a quince palabras, con un promedio de 5.5 palabras ambiguas por oración. En esta evaluación se reporta un 47% de precisión.

En [43] también se evalúo el algoritmo original de Lesk con templado simulado y *back-off* a sentido aleatorio sobre SENSEVAL-2, usando *WordNet* como diccionario. La ventana de contexto usada para los experimentos fue el tamaño de la oración y la medida de similitud usada fue el traslape normalizado, dividiendo entre largo de las definiciones, para evitar la ventaja de definiciones largas. Se reporta un 39.5% de precisión.

En [39] se evaluó el algoritmo de Lesk usando un algoritmo genético sobre un conjunto de 196 palabras en español obtenidas de un sitio de noticias de Internet. Como línea base de evaluación implementaron diferentes algoritmos, en todos los experimentos, el algoritmo genético mostró mejor precisión excepto por la línea base de sentido más frecuente.

6 Conclusiones

La desambiguación de sentidos de palabras es una de las tareas más importantes de la lingüística computacional o procesamiento del lenguaje natural, ya que tiene aplicación en muchas otras tareas de esta área.

Uno de los primeros algoritmos exitosos para llevar a cabo esta tarea fue el algoritmo original de Lesk, que se basa en dos ideas principales, un algoritmo de optimi-

zación y una medida de similitud para medir la relación entre las definiciones de los sentidos. El algoritmo de optimización considera la coherencia global del texto, esto es, encontrar la combinación de sentidos que maximice la relación total entre los sentidos de todas las palabras.

La principal ventaja de este algoritmo es que es un método no supervisado y sólo se necesita un diccionario de sentidos como recurso externo. Sus principales limitaciones son: con relación a la medida de similitud, que las glosas del diccionario regularmente son muy cortas y no incluyen el vocabulario suficiente para identificar los sentidos relacionados; y con relación al algoritmo de optimización, la explosión combinatoria que éste representa para un volumen de datos grande.

Por ello, diferentes métodos (conocidos como métodos tipo Lesk) han sido propuestos para aliviar estas limitaciones. En este artículo se presentó un estudio sobre estas propuestas y se presentó un análisis de los resultados que han sido obtenidos para las mismas.

Referencias

1. Bolshakov, I., Gelbukh, A.: Computational Linguistics. Models, Resources, Applications. Ciencia de la Computación. Primera Edición, México (2004)
2. Galicia-Haro, S.: Análisis sintáctico conducido por un diccionario de patrones de manejo sintáctico para lenguaje español. Tesis doctoral, CIC, IPN, México (2000)
3. Sidorov, G.: Etiquetador Morfológico y Desambiguador Manual: Dos Aplicaciones del Analizador Morfológico Automático para el Español. En: Memorias del VI encuentro internacional de computación ENC-2005, pp. 147–149, México (2005)
4. Gelbukh, A., Sidorov. G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, 240 pp. (2006).
5. Weaver, W.: Translation. Mimeographed, 12 pp., July 15, 1949. Reprinted in Locke, William N. y Booth, A. Donald (1955) (Eds.), Machine translation of languages. John Wiley & Sons, pp. 15-23, New York (1949)
6. Yngve, V.: Syntax and the problem of multiple meaning. In: Locke, William N. and Booth, A. Donald (Eds.), Machine translation of languages. John Wiley & Sons, pp. 208-226, New York (1955)
7. Salton, G.: Automatic Information organization and Retrieval. McGraw-Hill, New Cork (1968)
8. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
9. Krovetz, R., Croft, W.: Lexical Ambiguity and Information Retrieval. ACM Transactions on Information Systems, 10(2), pp. 115-141. (1992).
10. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval. Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 27 June-1 July 1993, pp. 171-180, Pittsburgh, Pennsylvania (1993)
11. Schütze, H., Pedersen, J.: Information retrieval based on word senses. Proceedings of SDAIR'95. Las Vegas, Nevada, Abril (1995).
12. Bolshakov, I., Gelbukh, A., Galicia-Haro, S.: A Simple Method to Detect and Correct Spanish Accentuation Typos. Proc. PACLING-99, Pacific Association for Computational Linguistics, Canada, pp. 104–113 (1999).

13. Yarowsky, D.: Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 88-95, Las Cruces, New Mexico (1994).
14. Hirst, G.: Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. WordNet An electronic Lexical Database. Edited by Christiane Fellbaum. The MIT Press. Cambridge, Massachusetts, London, England (1998)
15. Bolshakov, I., Gelbukh, A.: On Detection of Malapropisms by Multistage Collocation Testing. NLDB-2003, 8th International Conference on Application of Natural Language to Information Systems, Germany. Lecture Notes in Informatics. Bonner Köllen Verlag, pp. 28–41 (2003).
16. Bolshakov, I.A., Galicia-Haro, S.N., Gelbukh, A.: Detection and Correction of Malapropisms in Spanish by means of Internet Search. 8th International Conference Text, Speech and Dialogue (TSD-2005), Karlovy Vary, Czech Rep. Lecture Notes in Artificial Intelligence, N 3658, Springer, 2005, pp. 115–122 (2005).
17. Monroy, A., Calvo, H., Gelbukh, A.: NLP for Shallow Question Answering of Legal Documents Using Graphs. CICLing 2009. Lecture Notes in Computer Science N 5449, Springer, pp. 498–508 (2009).
18. Bhaskar, P., Pakray, P., Banerjee, S., Banerjee, S., Bandyopadhyay, S., Gelbukh, A.: Question Answering System for QA4MRE@CLEF 2012. CLEF 2012 Evaluation Labs and Workshop, Online Working Notes. Italy, 12 pp. (2012).
19. Lesk, M.: Automatic sense disambiguation using machine-readable dictionaries: how to tell a pine cone from an ice cream cone. Proc. of ACM SIGDOC Conference, p. 24–26, Toronto, Canada (1986)
20. Ledo Mezquita, Y., Gelbukh, A., Sidorov, G.: Recuperación de información con resolución de ambigüedad de sentidos de palabras para el español. Computación y Sistemas, vol. 11, no. 3, pp. 288–300 (2008).
21. Gelbukh, A., Sidorov, G., Chanona-Hernández, L.: Is word sense disambiguation useful in information retrieval? SSGRR 2003s, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet, track ISY, Scuola Superiore G. Reiss Romoli, L'Aquila, Italy (2003).
22. McCarthy, D., Koeling, R., Weeds, J., Carroll, J.: Finding predominant senses in untagged text. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain (2004)
23. Ríos Gaona, M.A., Godoy Calderón, S., Gelbukh, A.: Word Sense Disambiguation with the KORA-W Algorithm. Research in Computing Science, N 38, pp. 263–270 (2008).
24. Ríos Gaona, M.A., Gelbukh, A., Bandyopadhyay, S.: Web-based Variant of the Lesk Approach to Word Sense Disambiguation. Proc. of 2009 Eighth Mexican International Conference on Artificial Intelligence, IEEE CS Press, pp. 103–107 (2009).
25. Ledo Mezquita, Y., Sidorov, G., Gelbukh, A.. Tool for Computer-Aided Spanish Word Sense Disambiguation. CICLing-2003. Lecture Notes in Computer Science, N 2588, Springer, pp. 277–280 (2003).
26. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of ACL (1995).
27. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Computational linguistics and intelligent text processing, pp. 241-257, Springer Berlin Heidelberg (2003).
28. Banerjee, S., Pedersen, T.: An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February (2002)

29. Leacock, C., Chodorow, M., Miller, G.: Using corpus statistics and WordNet relations for sense identification. Computational Linguistics, 24(1), pp. 147-165 (1998)
30. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence,Montreal, August (1995)
31. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings on International Conference on Research in Computational Linguistics, Taiwan (1997)
32. Lin, D.: Using syntactic dependency as a local context to resolve word sense ambiguity. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pages 64–71, Madrid, July (1997)
33. Sinha, R., Mihalcea, R.: Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, September (2007)
34. Gelbukh, A., Sidorov, G., Han, S.-Y.: On Some Optimization Heuristics for Lesk-Like WSD Algorithms. Natural Language Processing and Information Systems. 10th International Conference on Applications of Natural Languages to Information Systems, NLDB-2005, Lecture Notes in Computer Science, N 3513, Springer, pp. 402–405 (2005).
35. Kilgarriff, A., Rosenzweig, J.: Framework and results for English SENSEVAL. Computers and the Humanities, 34 (1-2), (2000)
36. Cowie, L., Guthrie, J., Guthrie, L.: Lexical disambiguation using simulated annealing. COLING (1992)
37. Holland, J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975).
38. Goldberg D.E.: Genetic algorithms in search, optimization, and machina learning, Addison-Wesley, New York, (1989)
39. Gelbukh, A., Sidorov, G., Han, S.: Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization. WSEAS Transactions on Communications, Issue 1 Vol. 2, p. 11–19 (2003)
40. Gelbukh, A., Han, S.-Y., Sidorov, G.: Comparison of some global coherence optimization heuristics for word sense disambiguation. Avances en: Ciencias de la Computación, CIC-2003, XII Congreso Internacional de Computación, pp. 131–135 (2003).
41. Vasilescu, F., Langlais, P., Lapalme, G.: Evaluating variants of the Lesk approach for disambiguating words, LREC (2004)
42. Torres, S., Gelbukh, A.: Comparing similarity measures for original WSD lesk algorithm. Advances in Computer Science and Applications Research in Computing Science, 43, pp-155-166, México (2009)
43. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In Proceedings of HLT05, Morristown, NJ, USA (2005)

Aprendizaje de argumentos verbales completos y su plausibilidad en oraciones a partir de corpus[1]

Hiram Calvo

Centro de Investigación en Computación-IPN,
AV. Juan de Dios Bátiz S/N esq. M.O. de Mendizábal,
Col. Industrial Vallejo, México, D. F., 07738, MEXICO.

hcalvo@cic.ipn.mx

Resumen. El aprendizaje de preferencias de argumentos de verbos usualmente se ha tratado como un problema de verbo y argumento, o a lo mucho como una relación trinaria entre sujeto, verbo y objeto. Sin embargo, la correlación simultánea de todos los argumentos en una oración no ha sido explorado a profundidad para la medida de plausibilidad de una oración debido al alto número de combinaciones potenciales de argumentos, así como a la dispersión de los datos. En este trabajo presentamos una revisión de algunos métodos comunes para aprender las preferencias de los argumentos, comenzando con el caso más simple que considera correlaciones binarias, después lo comparamos con correlaciones trinarias, y finalmente consideramos todos los argumentos. Para esto último, usamos un modelo de aprendizaje en conjunto (*ensemble learning*) mediante modelos discriminativos y generativos; mediante características de coocurrencia y características semánticas en distintos arreglos. Buscamos responder preguntas acerca del número óptimo de tópicos requeridos para los modelos de PLSI y LDA, así como el número de coocurrencias que se requiere para mejorar el desempeño. Exploramos las implicaciones de usar diversas formas de proyectar correlaciones, es decir, en un espacio de palabras, o directamente en un espacio de coocurrencia de características. Realizamos pruebas para una tarea de pseudodesambiguación aprendiendo de corpus muy grandes extraídos de Internet.

1 Introducción

Una oración puede ser vista como un verbo con múltiples argumentos. La plausibilidad de cada argumento depende no sólo del verbo, sino también de otros argumentos. Medir la plausibilidad de los argumentos del verbo se requiere en diversas tareas como el etiquetado de roles semánticos, puesto que el agrupar los argumentos del verbo medir su plausibilidad incrementa el desempeño, tal como fue mostrado por Merlo y Van Der Plus (2009) y Deschacht y Moens (2009).

El reconocimiento de metáforas requiere esta información también, puesto que podemos conocer los usos comunes de los argumentos, y un uso no común podría sugerir la presencia de una metáfora, o un error de coherencia (por ejemplo *beber la luna en*

[1] Trabajo realizado con apoyo del CONACYT-SNI, y proyecto SIP-IPN.

un vaso). La detección de malapropismos puede usar también la medida de la plausibilidad de un argumento para determinar usos incorrectos de las palabras (Bolshakov, 2005), como en *centro histérico*, en lugar de *centro histórico*, *parece un tema* tattoo, *y hemos atrapado a dos personas* auspiciosas, *está entre la* espalda *y la pared*, etc. Por otra parte, la resolución de anáforas consiste en encontrar objetos referenciados, de tal manera que se requiere, entre otras cosas, tener información a la mano de la plausibilidad de los argumentos, es decir, qué tipo de palabra satisface las restricciones de la oración, como en: *El niño juega con* eso ahí, él *come pasto, y* lo *bebí en un vaso*.

Este problema puede ser visto como recolectar una base de datos grande de marcos semánticos con categorías detalladas y ejemplos que concuerdan con estas categorías. Para este propósito, existen diversos trabajos recientes que aprovechan los recursos manualmente manufacturados como WordNet, Wikipedia, FrameNet, VerbNet o PropBank. Por ejemplo, Reisinger y Paşca (2009) anotan conceptos existentes de WordNet con atributos, y extienden las relaciones de *es-un* basándose en el modelo de análisis latente de Dirichlet (LDA) en documentos web y la Wikipedia. Yamada y otros (2009) exploran la extracción de relaciones de hipónimos de Wikipedia usando descubrimiento basado en patrones, y agrupamiento de semejanza distribucional. El problema con el enfoque de marcos semánticos para esta tarea es que los marcos semánticos son demasiado generales.

Por ejemplo, Anna Korhonen (2000) considera los verbos *volar*, *navegar* y *resbalar* como similares, y encuentra un sólo marco de subcategorización. Por otra parte, los enfoques basados en n-gramas son demasiado particulares, e incluso usando un corpus muy grande (como la web como corpus) tiene dos problemas: algunas combinaciones no están disponibles, o las cuentas tienen sentencias hacia algunas estructuras sintácticas. Por ejemplo, resolver la adjunción de frase preposicional de *extinguir fuego con agua* usando Google da *fuego con agua:* 319,000 ocurrencias; *extinguir con agua*: 32,100, resultando en la estructura *(extinguir (fuego con agua)), en lugar de (extinguir (fuego) con agua). Es por ello que requerimos de algún mecanismo para ponderar estas cuentas. Esto último ha sido llevado a cabo mediante preferencias de selección desde Resnik (1996) para preferencias de verbo a clase, y después generalizado por Agirre y Martinez (2000) para preferencias de clases de verbos a clases de sustantivos.

Trabajos más recientes incluyen a McCarthy y Carroll (2003), que desambiguan sustantivos, verbos y adjetivos usando preferencias de selección aprendidas automáticamente como distribuciones de probabilidad sobre la jerarquía de hipónimos de los sustantivos de WordNet, evaluando con Senseval-2. Sin embargo, estos trabajos mencionados tienen un problema en común, y es que consideran por separado cada argumento para un verbo.

1.1 Un argumento no es suficiente

Considere la siguiente oración:

Hay alfalfa en la granja. La vaca la come.

Quisiéramos conectar “la” con “alfalfa”, y no con “granja”. A partir de las preferencias de selección sabemos que el objeto de *comer* debería ser algo comestible, de tal

forma que sabemos que *alfalfa* es más comestible que *granja*, resolviendo este problema. A partir de marcos semánticos tenemos conocimiento similar, pero en un sentido más amplio: hay un Investor y un ingestible.

Sin embargo, esta información puede ser insuficiente en algunos casos cuando la preferencia de selección depende de otros argumentos de la frase. Por ejemplo:

La vaca come alfalfa, pero el hombre la comerá.

En este caso, no es suficiente con saber qué objeto es comestible, sino que la resolución depende de quién está comiendo. En este caso es improbable que el hombre coma alfalfa, así que la oración podría referirse al hecho de que él comerá a la vaca. Esto mismo ocurre con otros argumentos para verbos. Por ejemplo, algunos de los argumentos periféricos de FrameNet para el marco de *ingestión* son *instrumento* y *lugar*. Sin embargo, existen algunas cosas que se comen con un instrumento en particular, por ejemplo, la sopa se come con una cuchara, mientras que el arroz se come con un tenedor o con palillos, dependiendo de quién come, o el lugar donde se come. La extracción de argumentos plausibles permite construir un diccionario que funja como base de datos para este tipo de información, que puede ser vista a su vez como sentido común, puesto que es posible aprender qué tipo de actividades son desarrolladas por grupos o entidades automáticamente a partir de grandes bloques de texto.

El objetivo de nuestro trabajo es construir dicha base de datos. Para este propósito necesitamos obtener información relacionada con las preferencias de selección y la extracción de marcos semánticos.

En las siguientes secciones presentaremos el trabajo relacionado, organizado en diversos enfoques (sección 2), después presentaremos una propuesta basada en el modelo de espacio vectorial (sección 3), después una propuesta basada en modelos de lenguaje (sección 4), y finalmente presentamos nuestros aportes principales (secciones 5 y 6), que consisten en aplicar indexado probabilístico semántico latente (PLSI) para manejar tres variables correlacionadas (sección 5), y finalmente el manejo de la coocurrencia compleja mediante máquinas de soporte vectorial (SVM) a partir de características provistas por PLSI y coocurrencia (sección 6). En cada sección presentaremos diversos experimentos para mostrar cómo diferentes parámetros afectan el comportamiento del modelo, así como para comparar diversos enfoques.

1.2 Enfoques para aprender preferencias de argumentos para los verbos

El problema de aprender la plausibilidad de los argumentos de un verbo puede ser estudiado desde diversos puntos de vista. Desde el punto de vista del tipo de información extraída, podemos encontrar trabajos relacionados para preferencias de selección y extracción de marcos semánticos. Desde el punto de vista de las preferencias de selección, la tarea se enfoca en obtener automáticamente clases de argumentos para un verbo y una construcción sintáctica dados. Desde el enfoque de marcos semánticos, los argumentos se agrupan por el rol semántico que tienen, sin importar la construcción sintáctica que tengan. Este último enfoque enfatiza la distinción entre argumentos indispensables (núcleo) y periféricos. Por otra parte, podemos considerar el punto de vista de cómo esta información se representa: la tarea puede ser vista como un caso de modelado

estadístico del lenguaje, donde, dado un contexto (verbo y otros argumentos), el argumento faltante debe ser inferido con una alta probabilidad; o puede ser observado como una tarea de modelo de espacio de palabras frecuentemente visto en sistemas de recuperación de información. En las siguientes secciones presentamos trabajos relacionados a esta tarea desde estos distintos puntos de vista.

1.2.1 Preferencias de selección

La adquisición de preferencias de selección puede verse como uno de los primeros intentos para encontrar automáticamente la plausibilidad de los argumentos. Los intentos tempranos trataban con pares simples de verbo y argumento. Puesto que el recurso de aprendizaje es vasto y disperso, todos estos trabajos utilizan un mecanismo de generalización, o suavizado, para extender la cobertura. Resnik (1996) utiliza WordNet para generalizar el argumento de tipo objeto. Agirre y Martinez (2001) usan un modelo de clase a clase, de tal forma que tanto el verbo como el argumento objeto se generalizan al pertenecer a una clase usando WordNet. McCarthy y Carroll (2006) obtienen preferencias de selección como distribuciones probabilísticas aparte del argumento objeto. Padó y Lapata (2007) combinan información semántica y sintáctica estimando su modelo usando corpus con anotación de roles semánticos (por ejemplo FrameNet, PropBank), y después aplicando suavizado basado en clases mediante WordNet.

1.2.2 Marcos de subcategorización

Los siguientes trabajos tratan el problema de la extracción de la plausibilidad de argumentos de forma semisupervisada desde el enfoque de la extracción de marcos de subcategorización. Salgeiro *et al*. obtienen estructuras de argumentos de verbos. Generalizan sustantivos usando un reconocedor de entidades nombradas (IdentiFinder) y después utilizan el entorno del canal ruidoso para predecir argumentos. Ejemplos del tipo de información con la que trabajan son: *organización* compró *organización* de *organización*. *Cosa* compró las acciones en *fecha*, y a veces sin generalización, *La cafetería compró platos extras.*

Otro trabajo semisupervisado es el de Kawahara y Kurohashi (2001). Ellos generalizan utilizando un diccionario de ideas afines manualmente creado. Para encontrar los marcos de casos, usan junto con el verbo el argumento más cercano, proveyendo de desambiguación del sentido del verbo para casos similares al ejemplo que nos motivó, presentado en la sección 1.

A continuación presentaremos otros puntos de vista que tratan con la representación de la información de los argumentos del verbo.

1.2.3 El modelo de espacio de palabras, o modelo de espacio vectorial

Tradicionalmente según los modelos de recuperación de información, las palabras pueden representarse como documentos, y los contextos semánticos como características,

de tal forma que es posible construir una matriz de coocurrencia, o un espacio de palabras, donde cada intersección de palabra y contexto muestra el conteo de la frecuencia de aparición. Este enfoque ha sido usado recientemente con relaciones sintácticas (Padó y Lapata, 2007). Una cuestión importante dentro de este enfoque es la medida de semejanza elegida para comparar palabras (documentos) dadas sus características. Las medidas de semejanza comunes van desde medidas simples como la medida euclidiana, la medida coseno, y el coeficiente de Jaccard (Lee, 1999), hasta medidas como la medida de Hindle y la medida de Lin.

1.2.4 Modelo del lenguaje

Podemos ver la tarea de encontrar la plausibilidad de cierto argumento para un conjunto de oraciones como estimar una palabra dado un contexto específico. Particularmente, para este trabajo podemos considerar el contexto como las relaciones gramaticales para un verbo en particular:

$$P(w,c) = P(c) \cdot P(c|w)$$

que puede ser estimada de muchas formas. Particularmente, usando un modelo oculto de Markov, o utilizando variables latentes para el suavizado, como ya vimos con los modelos probabilísticos de indexado semántico latente (PLSI) (Hoffmann, 1999):

$$P(w,c) = \sum_{z_i} P(z) \cdot P(w|z) \cdot P(c|z)$$

La probabilidad condicional puede ser calculada a partir de conteos de frecuencia de n-gramas.

En las siguientes secciones presentaremos una propuesta simple dentro del enfoque del modelo de espacio de palabras (sección 2); posteriormente presentaremos dos algoritmos dentro del enfoque de modelo del lenguaje (sección 3).

2 Un modelo de espacio de palabras

Comenzaremos con un modelo simple para explorar las posibilidades de los últimos dos enfoques. En esta sección proponemos un modelo basado en el modelo de espacio de palabras.

Para el modelo de espacio de palabras, podemos construir una matriz donde a_2 son los renglones (documentos) y v, a_1 son características. Puesto que esta matriz es muy dispersa, usamos un diccionario de ideas afines para suavizar los valores de los argumentos. Para hacer esto, seguimos libremente el enfoque propuesto por (McCarthy *et al.*, 2004) para encontrar el sentido más frecuente, pero en este caso usamos los k vecinos más cercanos a cada argumento a_i para encontrar el predominio de una tripleta no vista dada su semejanza a todas las tripletas presentes en el corpus, midiendo la semejanza entre argumentos. En otras palabras, como en (McCarthy *et al.*, 2004, Tejada *et*

al., 2008a, 2008b) para desambiguación de los sentidos de las palabras, cada argumento semejante vota por la plausibilidad de cada tripleta.

$$Predominio(V, X_1, X_2) = \frac{\sum_{<v,a_1,a_2>\in T} sim(a_1, x_1) P_{MLE}(v, a_1, a_2)}{\sum_{<v,a_1,a_2>\in T} sim\_existe(a_1, a_2, x_1, x_2)}$$

donde T es el conjunto completo de tripletas $<$*verbo*, *argumento*$_1$, *argumento*$_2$$>$, P_{MLE} es la máxima verosimilitud de $<$*verbo*, *argumento*$_1$, *argumento*$_2$$>$, y

$$sim\_existe(a_1, a_2, x_1, x_2) = \begin{cases} 1 & \text{si } sim(a_1, x_1) \cdot sim(a_2, x_2) > 0 \\ 0 & \text{de otra forma} \end{cases}$$

Para medir la semejanza entre argumentos construimos un diccionario de ideas afines usando el método descrito por Lin (1998a) usando el analizador sintáctico Minipar (Lin, 1998b) sobre relaciones de corta distancia; es decir, previamente habíamos separado las oraciones subordinadas. Obtuvimos tripletas $<v,a_1,a_2>$ a partir de este corpus, que fueron contadas, y éstas fueron utilizadas tanto para construir el tesauro, como para ser utilizadas como fuente de coocurrencias de verbos y argumentos.

2.1 Evaluación

Comparamos estos dos modelos en una tarea de pseudodesambiguación siguiendo a Weeds y Weir (2003). Primero, obtuvimos tripletas $\langle v, a_1, a_2 \rangle$ del corpus. Después, dividimos el corpus en entrenamiento (80%) y prueba (20%). Con la primera parte entrenamos el modelo probabilístico de indexado semántico latente y creamos el modelo de espacio de palabras. Este modelo de espacio de palabras también se utilizó para obtener la medida de semejanza para cada par de argumentos. De esta forma podremos calcular la plausibilidad de $\langle v, a_1, a_2 \rangle$. Para la evaluación creamos 4-tuplas artificialmente: $\langle v, a_1, a_2, a_2' \rangle$, formadas al tomar todas las tripletas $\langle v, a_1, a_2 \rangle$ del corpus de prueba, y generando una tupla artificial $\langle v, a_1, a_2' \rangle$ eligiendo una a_2' aleatoria tal que $r_2' = r_2$, asegurándose de que esta nueva tripleta $\langle v, a_1, a_2 \rangle$creada aleatoriamente no estuviera presente en el corpus de entrenamiento. La tarea consiste en seleccionar la tupla correcta. Es posible que ocurran empates cuando ambas tuplas tienen la misma calificación (y ambas son distintas de cero). Comparamos los dos modelos, uno basado en modelos estadísticos del lenguaje (vea la sección 3) y el modelo de espacio de palabras. Utilizando el corpus de patentes de la colección NII de prueba para el sistema de recuperación de información NTCIR-5 (Fuji and Iwayama, 2005), analizamos 7,300 millones de palabras, y después extrajimos la cadena de relaciones de una forma dirigida, es decir, para la oración: X suma Y a Z por W, extrajimos las tripletas <suma, suj-X, obj-Y>, <suma, obj-Y, a-Z>, y <suma, a-Z, por-W>. Obtuvimos 706 millones de tripletas de la forma <*v*, *a1*, *a2*>. Consideramos sólo relaciones asimétricas encadenadas para evitar semejanzas falsas entre palabras que coocurren en la misma oración.

Siguiendo a Weeds y Weir (2003), elegimos 20 verbos, cubriendo verbos de alta frecuencia y verbos de baja frecuencia, y para cada uno extrajimos todas las tripletas $<v, a_1, a_2>$ presentes en el corpus de tripletas. Después realizamos los experimentos con el algoritmo basado en PLSI y el algoritmo basado en el modelo de espacio de palabras (WSM).

Experimentamos con diferentes números de tópicos para la variable latente *z* en PLSI, y con un número diferente de vecinos para el tesauro de Lin para expandir el modelo de espacio de palabras. Los resultados se muestran en la figura 2.

2.2 Análisis

Hemos mostrado resultados para un algoritmo basado en el enfoque del modelo de espacio de palabras para la extracción no supervisada de argumentos plausibles para un verbo, y lo comparamos con un enfoque probabilístico de indexado semántico latente (PLSI), encontrando evidencia particular para respaldar la afirmación de que es posible lograr buenos resultados con el método que vota por tripletas comunes usando un tesauro distribucional. Los resultados parecen ser consistentes con trabajos previos que usan diccionarios de ideas afines (Calvo *et al.*, 2005; Tejada *et al*., 2008a; 2008b): el añadir información incrementa la cobertura con poco sacrificio en cuanto a precisión.

No usamos ningún otro recurso después del analizador de dependencias, como reconocedores de entidades nombradas, o datos etiquetados para entrenar a un algoritmo de aprendizaje por computadora, así que a partir de esta etapa, el algoritmo es no supervisado.

Para desarrollar más este enfoque, es necesario experimentar con el límite superior del incremento de cobertura, puesto que cada vecino del diccionario de ideas afines está

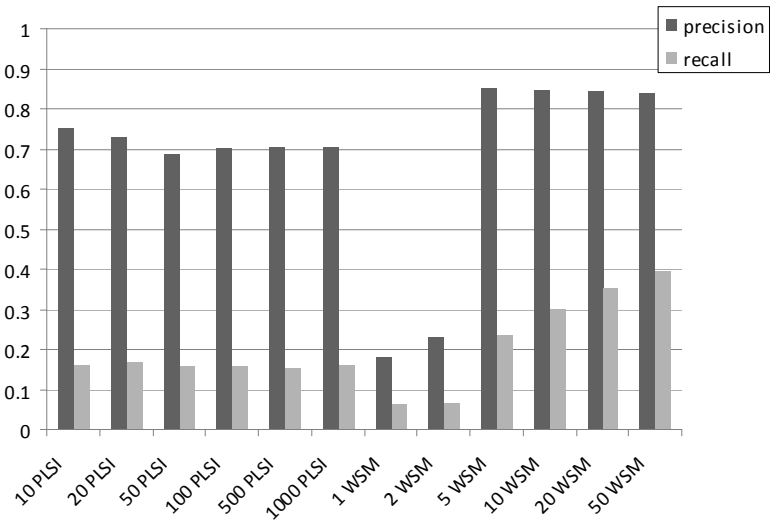

Figura 1. Resultados para (tópicos)-PLSI y (vecinos)-WSM.

añadiendo ruido. Hemos experimentado con la construcción del tesauro usando el mismo corpus; sin embargo, podrían encontrarse diferencias significativas si se usa un corpus enciclopédico para construir el diccionario, pues podría contarse con un contexto más amplio y rico.

Como trabajo futuro, también es posible experimentar con el efecto de usar otras medidas de semejanza, así como construir una tabla de semejanzas con objetos más simples: un sustantivo simple en lugar de un objeto compuesto.

En la siguiente sección exploraremos propuestas que se encuentran dentro del modelo del lenguaje.

3 Modelo del lenguaje basado en dependencias

La mayor parte del trabajo previo en modelos del lenguaje estadísticos está orientado a tareas de reconocimiento de voz (Clarkson and Rosenfeld, 1997; Rosenfeld, 2000) mediante modelos de entropía máxima. Usualmente, debido a limitaciones de espacio estos modelos se limitan a modelos secuenciales de trigramas. Diversos trabajos (Gao y Suzuki, 2003; Gao *et al.*, 2004) han mostrado que depender únicamente de n-gramas secuenciales no es siempre la mejor estrategia. Considere el ejemplo tomado de (Gao y Suzuki, 2003): *[Un bebé] [en el asiento de al lado] lloró [durante todo el vuelo]*. Un modelo de n-gramas trataría de predecir lloró a partir de *al lado*, en tanto que un modelo del lenguaje basado en dependencias (DLM por sus siglas en inglés) trataría de predecir *lloró* a partir de *bebé*.

En esta sección exploramos la creación de un DLM para obtener ocupantes de escenarios factibles, que pueden ser vistos como extraer preferencias de selección (Resnik, 1996) pero con un contexto más amplio para cada ocupante. Mostramos en la sección 3.1.1 cómo esta información adicional ayuda a obtener el mejor candidato ocupante; posteriormente en la sección 3.1.2 y 3.1.3 presentamos nuestras implementaciones de dos modelos para crear un DLM, uno basado en modelos probabilísticos de indexado semántico latente (PLSI) (sección 3.1.2) y uno basado en los k vecinos más cercanos (KNN) (sección 3.1.3). En la sección 3.2 describimos nuestros experimentos para comparar ambos algoritmos en una tarea de pseudodesambiguación. Analizaremos nuestros resultados en la sección 3.3.

3.1 Modelos para la estimación de argumentos plausibles

3.1.1 Ocupantes factibles para escenarios

Consideremos que queremos encontrar el objeto más factible de ser comido dado del verbo *comer*. Puesto que *comer* tiene diversos sentidos, el ocupante para el rol de objeto comido de *comer* podría ser comida, o podría ser un material, dependiendo de quién está comiendo. Por ejemplo, si el sujeto es ácido, entonces el objeto comido podría ser *metal*, o algún otro material (el ácido se *come* al metal).

Si se considera el problema de estimar $P(a_2|v,a_1)$ en lugar de estimar únicamente $P(a_2|v)$, donde a_1 y a_2 son argumentos, y v es un verbo, puede verse que el problema

de la dispersión de los datos se aumenta. Esto ha sido resuelto principalmente usando recursos externos como WordNet (Resnik, 1996; McCarthy and Carroll, 2006; Agirre and Martinez, 2001); recursos anotados con roles semánticos, como FrameNet, PropBank (Padó and Lapata, 2007); un reconocedor de entidades nombradas como IdentiFinder (Salgeiro *et al.*, 2006); u otros diccionarios de ideas afines manualmente creados (Kawahara and Kurohashi, 2001).

Un objetivo de esta sección es encontrar hasta qué grado la información del corpus en sí mismo puede ser utilizada para estimar $P(a_2|v,a_1)$ sin utilizar recursos adicionales. Para esto, diversas técnicas se utilizan para tratar con el problema de dispersión de los datos. Describimos dos de ellas en la siguiente sección.

3.1.2 PLSI – Modelo probabilístico de indexado semántico latente

Puesto que queremos considerar la correlación de los argumentos, usaremos la siguiente información: $P(v,r_1,n_1,r_2,n_2)$, donde *v es un verbo*, r_1 es la relación entre el verbo y n_1 (sustantivo) como sujeto, objeto, preposición o adverbio. r_2 y n_2 son análogos. Si asumimos que *n* tiene una función diferente cuando se usa con otra relación, entonces podemos considerar que *r* y *n* forman un nuevo símbolo, llamado *a*. De esta forma podemos simplificar nuestra 5-tupla a $P(v,a_1,a_2)$. Queremos saber, dado un verbo y un argumento a_1, cuál a_2 es el más plausible, es decir, queremos saber $P(a_2|v,a_1)$. Podemos escribir la probabilidad de encontrar u verbo en particular y dos de sus relaciones sintácticas como:

$$P(v,a_1,a_2) = P(v,a_1)\ P(a_2|v,a_1),$$

que puede ser estimada de distintas formas. Particularmente para este trabajo, usamos el modelo probabilístico de indexado latente semántico (Hoffmann, 1999) porque podemos explotar el concepto de variables latentes que se encargan de la dispersión de los datos.

El modelo probabilístico de indexado latente semántico (PLSI por sus siglas en inglés) fue introducido en (Hofmann, 1999), y surgió del indexado latente semántico (Deerwester et al., 1990). Este modelo intenta asociar una variable de clase no observada $z \in Z=\{z_1, ..., z_k\}$, (en nuestro caso una generalización de la correlación de la coocurrencia de v,a_1 y a_2), y dos conjuntos de observables: argumentos, y verbos+argumentos. En términos de un modelo generativo puede ser definido como sigue: se selecciona un par v,a_1 con probabilidad $P(z|v,a_1)$ y finalmente un argumento a_2 es seleccionado con probabilidad $P(a_2|z)$. Usando PLSI según (Hoffmann, 1999), es posible obtener:

$$P(\mathrm{v}, a_1, a_2) = \sum_z P(z_i)P(a_2|z_i)P(\mathrm{v}, a_1|z_i),$$

donde *z* es una variable latente que captura la correlación entre a_2 y la coocurrencia de (v,a_1) simultáneamente. Usando una variable latente para correlacionar tres variables puede conducir a un mal desempeño de PLSI, por lo que en la siguiente función exploraremos diversas formas de explotar el suavizado por variables latentes semánticas.

3.1.3 Modelo de K vecinos más cercanos (KNN-expansor)

Este modelo usa los *k* vecinos más cercanos de cada argumento para encontrar la plausibilidad de una tripleta no vista, dada su semejanza con todas las tripletas presentes en el corpus, midiendo su semejanza entre argumentos. Puesto que los votos son acumulativos, las tripletas que tienen palabras con muchas palabras semejantes tendrán más votos.

Las medidas usuales de semejanza incluyen la distancia euclidiana, coseno, y el coeficiente de Jaccard. Weeds y Weir (2003) muestran que la medida de semejanza con mejor desempeño es la medida distribucional de Lin, así que usamos esta medida para suavizar a los K vecinos más cercanos, siguiendo el procedimiento descrito por (Lin, 1998b).

3.2 Experimentos y evaluación

Para estos experimentos, usamos el mismo marco presentado en la sección 2.1. Creamos 4-tuplas artificiales $\langle v, a_1, a_2, a'_2 \rangle$, formadas tomando todas las tripletas $\langle v, a_1, a_2 \rangle$ del corpus de prueba, y generando una tripleta artificial $\langle v, a_1, a'_2 \rangle$ eligiendo una a'_2 aleatoria con $r'_2 = r_2$, asegurándose de que esta nueva tripleta aleatoria $\langle v, a_1, a'_2 \rangle$ no estuviera presente en el corpus de entrenamiento. La tarea consiste en seleccionar la tripleta correcta. Al igual que en la sección 2.1, utilizamos el corpus NTCIR-5 Patent.

3.2.1 Comparación del efecto de añadir contexto

Para este experimento, creamos un mini-corpus conjunto consistente en 1000 tripletas para cada uno de ciertos verbos elegidos del corpus de patentes: añadir, calcular, venir, hacer, comer, fijar, ir, tener, inspeccionar, aprender, gustar, leer, ver, parecer y escribir). Queremos evaluar el impacto de añadir más información para la predicción de los argumentos de los verbos, así que estimamos la plausibilidad de un argumento dado un verbo: $P(a_2|v)$; después la comparamos con el uso de información adicional de otros argumentos para ambos modelos: $P(a_2|v,a_1)$.

Para palabras completamente nuevas a veces no es posible tener un estimado, así que medimos tanto precisión como recuperación. La precisión mide cuántas adjunciones se predijeron correctamente de los ejemplos cubiertos, mientras que la recuperación mide la adjunción correcta para todo el conjunto de prueba. Nos interesa en medir la precisión y la recuperación de estos métodos, así que no implementamos ninguna técnica de retroceso.

3.3 Análisis

La operación por separado en verbos (un mini-corpus por verbo) da mejores resultados para PLSI (la precisión se encuentra arriba de 0.8), sin embargo esto parece no afectar

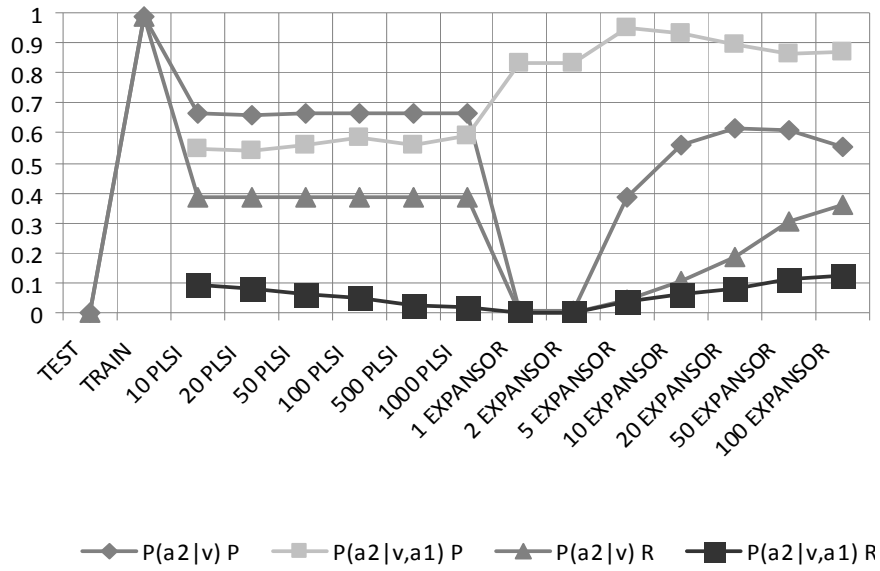

Figura 2. Efecto de añadir más contexto: predicción basa sólo en el verbo vs. predicción basada en el verbo + un argumento. KNN-Expansor es un modelo basado en *k* vecinos más cercanos

a KNN-Expansor. Para poco contexto $P(a_2|v)$, PLSI funciona mejor que KNN-Expansor. Para más contexto, $P(a_2|v,a_1)$, KNN-Expansor funciona mejor.

En general, PLSI prefiere un número pequeño de tópicos, incluso para un corpus grande (al rededor de 20 tópicos para el corpus más grande de experimentos). KNN-Expansor parece mejorar la recuperación uniformemente cuando se añaden más vecinos, perdiendo poca precisión. Expandir con pocos vecinos (1 a 5) parece no ser muy útil. Particularmente es posible ver en la Figura 2 que cuando la recuperación es muy baja, la precisión puede ser muy alta o muy baja. Esto es porque cuando se resuelven muy pocos casos, el desempeño prácticamente tiende a ser aleatoria. En general, los resultados de recuperación parecen ser bajos debido a que no utilizamos ningún método de retroceso. Si comparamos la precisión de KNN-Expansor, modelo completo (basado en más contexto), podríamos pensar que retroceder a PLSI basado en pares $P(a_2|v)$ daría mejores resultados, pero esto se ha dejado como trabajo futuro.

Evaluamos dos diferentes modelos del lenguaje basados en dependencias con una prueba de pseudodesambiguación. El modelo basado en vecinos cercanos (KNN-Expansor) se comporta mejor que PLSI cuando se incrementa la dispersión de los datos al añadir más información. Un suavizado efectivo se logra al votar usando medidas de semejanza del tesauro distribucional de Lin.

Puesto que el modelo PLSI que estamos usando tiene que manejar diversos argumentos con una sola variable latente, es posible pensar en una mejora que consiste en

interpolar diversos modelos de PLSI para manejar diversos argumentos. En la siguiente sección daremos detalles de este modelo.

4 PLSI interpolado

En esta sección proponemos un nuevo modelo llamado PLSI interpolado, que permite usar múltiples variables semánticas latentes. Este algoritmo está basado en el algoritmo descrito en la sección 3.1.2.

4.1 iPLSI – PLSI interpolado

La fórmula para PLSI previamente utilizada aglomera la asociación de información de a_2 y v, a_1 simultáneamente en una misma variable latente. Esto causa dos problemas: primero, escasez de los datos, y segundo, fija la correlación entre dos variables. De aquí que propongamos una variación para este cálculo usando interpolación basada en cada par de argumentos para una tripleta.
Una forma interpolada para estimar la probabilidad de una tripleta basada en las coocurrencias de sus diferentes pares está dada por:

$$P_E(v,a_1,a_2) \approx f_m(v,a_1) f(a_2) + f_n(v,a_2) f(a_1) + f_o(a_1,a_2) f(a_2) + f_a(v,a_1,a_2) + f_b(v,a_1,a_2) + f_c(v,a_1,a_2)$$

$$f_a(v,a_1,a_2) = \sum_a P(a_i) \cdot P(v,a_2|a) \cdot P(a_1|a)$$

$$f_b(v,a_1,a_2) = \sum_b P(b_i) \cdot P(a_1,a_2|b_i) \cdot P(v|b_i)$$

$$f_c(v,a_1,a_2) = \sum_c P(c_i) \cdot P(v,a_1|c_i) \cdot P(a_2|c_i)$$

Note que a_i (los tópicos de la variable latente) no debe ser confundida con a_1 y a_2 (los argumentos).

4.2 Experimentación

Comparemos estos dos modelos en una tarea de desambiguación, como se mostró en la sección 2.1 y 3.2. Sin embargo, para tener un rango más amplio de palabras coocurrentes, para estas evaluaciones utilizamos el corpus UKWaC (Ferraresi et al. 2008). Este corpus es un corpus grande balanceado tomado de la web, con más de 2 billones de

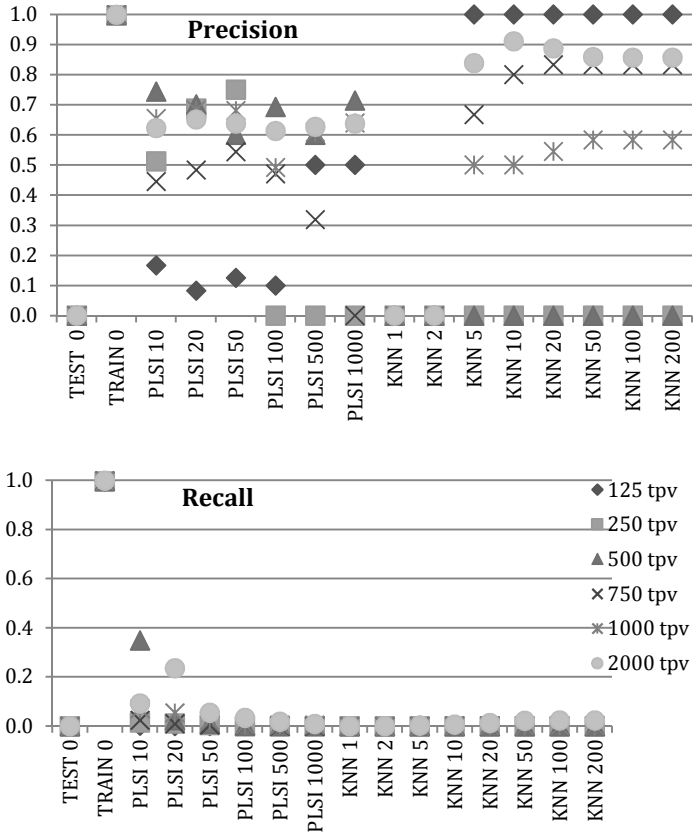

Figura 3. Promedio de precisión y cobertura para los originales PLSI y KNN-Expansor, mostrando la tasa de aprendizaje (cada serie tiene un número diferente de tripletas por verbo, tpv). No se usó umbral de frecuencia. Los números en la parte inferior muestran el número de tópicos para PLSI y el número de vecinos para KNN.

palabras[2]. Creamos dos conjuntos de palabras para los verbos: jugar, comer, sumar, calcular, fijar, leer, escribir, tener, aprender, inspeccionar, gustar, hacer, venir, ir, ver, parecer, dar, tomar, mantener, hacer, poner, enviar, decir, obtener, caminar, correr, estudiar, necesitar y devenir. Estos verbos fueron elegidos como una muestra de verbos frecuentes, y verbos poco frecuentes.

También hay verbos que pueden tomar una gran variedad de argumentos, como *tomar* (es decir, su ambigüedad es alta). Cada conjunto de palabras contiene 1000 o 2500 tripletas de verbos para cada verbo. El primer conjunto de palabras se evaluó contra 5,279 tripletas de dependencias de verbos, mientras que el segundo conjunto de palabras se evaluó con 12,677 tripletas de dependencias de verbos, correspondiendo aproximadamente al 20% del total de tripletas en cada conjunto de palabras.

[2] Una herramienta para hacer consultas de concordancias a este corpus puede encontrarse en http://sketchengine.co.uk

4.2.1 Resultados del algoritmo original con el nuevo corpus

En esta sección presentamos nuestros resultados para el nuevo corpus presentado en la sección 4.2. Las pruebas fueron llevadas a cabo con un 7 tópicos para PLSI y un valor de 100 vecinos para KNN-Expansor. Vimos en la sección 3.2 que para estimar la probabilidad de un argumento a_2, $P(a_2|v,a_1)$ funciona mejor que $P(a_2|v)$. Los experimentos realizados con este nuevo corpus confirman esto para distintos tamaños de los conjuntos de palabras. En la mayoría de los casos KNN-Expansor se comporta mejor que el PLSI original en precisión y recuperación (la mejor de las variaciones de KNN-Expansor es mejor que la mejor de las variaciones de PLSI). Contrario a KNN-Expansor, el desempeño de PLSI se incrementa en tanto que se incrementa el tamaño del conjunto de palabras, probablemente debido a que existe mayor confusión al usar el mismo número de tópicos. Esto puede ver también en la Figura 3: la recuperación mejora ligeramente para conjuntos de data mayores, y con más tópicos.

4.2.2 Medición de la tasa de aprendizaje

Este experimento consistió en incrementar gradualmente el número de tripletas de 125 a 2000 tripletas de dependencias por verbo (tpv) para examinar los efectos de usar corpus más pequeños. Los resultados se muestran en la Figura 3. En esta figura KNN-Expansor sobrepasa a PLSI cuando se añade más datos. La precisión de KNN es más alta también en general en todos los experimentos. Los mejores resultados para PLSI se obtuvieron con 7 tópicos, mientras que para KNN los mejores resultados se obtuvieron con 200 vecinos.

4.2.3 Resultados sin pre-filtrado

Los resultados anteriores usaban un umbral de pre-filtrado de 4, esto quiere decir que tripletas con menos de 4 ocurrencias fueron desechadas. Cuando quitamos este filtro, los resultados para KNN caen dramáticamente. PLSI es capaz de mantener un buen desempeño con 20 tópicos. Esto sugiere que PLSI es capaz de suavizar mejor ocurrencias simples para ciertas tripletas. KNN es mejor al trabajar con tripletas que ocurren frecuentemente. Requerimos un método que puede manejar ocurrencias de palabras no frecuentes, puesto que el pre-filtrado implica cierta pérdida de información que podría ser útil posteriormente. Por ejemplo, imagine que *tezgüino* se menciona sólo una vez en el conjunto de entrenamiento. Consideramos que es importante poder aprender información de entidades mencionadas escasamente. La siguiente sección presenta resultados con respecto a la mejora de PLSI para manejar elementos no filtrados.

4.3 Resultados de iPLSI

Como vimos en la sección 4.1, probamos diferentes modelos para combinar las variables semánticas latentes. El mejor modelo que obtuvimos combina las medidas de

(v,a_1), (a_1,a_2) y (v,a_2), respectivamente, dando una precisión de 0.83 y una recuperación de 0.83.

También realizamos pruebas con n-gramas puros (sin utilizar tripletas de dependencias, como en todas las pruebas anteriores). Veremos que los mismos componentes también dan la mejor solución.

4.4 Prueba con n-gramas

Realizamos esta prueba para corroborar que los tres componentes están contribuyendo a la interpolación, así como para evitar la tendencia que el analizador sintáctico pudiera estar provocando.

La prueba de n-gramas se realizó seleccionando trigramas de bigramas del corpus UKWaC de una forma parecida a la de los experimentos previos. Sin embargo, en este caso no usamos relaciones de dependencias, sino ventanas deslizantes de hexagramas distribuidas en trigramas como un intento de imitar la forma en la que palabras de función (como preposiciones o determinantes) afectan a las tripletas en el modelo de dependencias. Los trigramas fueron extraídos para n-gramas relacionados con los mismos verbos descritos en la sección 4.2.

La tarea consistió, como con la tarea de las tripletas de dependencias, en elegir una entre dos opciones del par 1. Usamos 80% de los trigramas como una base para entrenamiento, y el 20% para la prueba. Las pruebas se realizaron con 500 tripletas por verbo hasta 5,000 tripletas por 2000, probando todas las combinaciones posibles de elementos de (v, a_1, a_2). Al igual que en el caso anterior, combinar las medidas de (v,a_1), (a_1,a_2) y (v,a_2) tuvo el mejor desempeño (precisión de 0.77 y 0.77 de recuperación para 500 tripletas por verbo).

4.5 Análisis

Vimos que el algoritmo KNN-Expansor se desempeña mejor que el PLSI de una sola variable latente, y estudiamos la tasa de aprendizaje de ambos algoritmos, mostrando que KNN incrementa la recuperación cuando se le añaden más datos, sin perder mucha precisión; sin embargo, KNN-Expansor requiere fuertemente una fase de pre-filtrado que eventualmente conduce a una pérdida importante de palabras que ocurren escasamente.

Estas palabras son importantes para nuestros propósitos, pues quitarlas nos quita la posibilidad de generalizar palabras raras para medir su plausibilidad. El algoritmo propuesto de PLSI interpolado (iPLSI) soluciona este problema, dando mejores resultados que el PLSI de una sola variable. Encontramos que es posible seleccionar el hexagrama más factible de dos, con un 77% de recuperación para n-gramas puros agrupados como trigramas de bigramas, y hasta un 83% de recuperación para trigramas de dependencias.

Las pruebas conducidas muestran que es posible seleccionar el candidato correcto para una tripleta que puede ser vista como parte de una oración. Esto permite calcular el argumento más plausible en una oración, usando un contexto más amplio dado por un verbo y otro argumento.

iPLSI ha funcionado mejor que el modelo previo de KNN, pero aún quedan aspectos para mejorar. Particularmente, estamos estimando la coocurrencia de dos argumentos simultáneamente. Para determinar si usar más argumentos es mejor para la predicción de argumentos, proponemos un modelo que nos permite hacer esto en la siguiente sección, y después lo comparamos con los métodos previos.

5 La necesidad de medir todas las coocurrencias

Hemos visto previamente que considerar simultáneamente tres argumentos da mejor precisión que considerar únicamente dos, con cierta pérdida de recuperación. Kawahara y Kurohashi (2006) realizan desambiguación de verbos para aprender preferencias diferenciando el verbo principal con el argumento más cercano. Por ejemplo *jugar una broma* y *jugar un juego* tendrán distintas preferencias de sus otros argumentos; sin embargo, en algunos casos esto no es suficiente, como puede verse en el siguiente ejemplo, donde el verbo tiene diferentes significados dependiendo de un argumento lejano:

Poner una escena para los amigos en el teatro (montar, actuar) y
Poner una escena para los amigos en la TV (reproducir)

Trabajos recientes han propuesto un enfoque discriminativo para aprender las preferencias de selección, comenzando con Bergsma *et al.* (2008). Ritter *et al.* (2010) y Ó Séaghdha (2010) proponen LinkLDA (Latent Dirichlet Allocation), un modelo con variables de tópicos ocultas obtenidas de la misma distribución para modelar combinaciones de <sujeto, verbo, objeto> tales como <hombre, come, ramen> y <vaca, come, pasto>.

Sin embargo, estos trabajos consideran a lo más relaciones trinarias. Motivados por el problema de considerar tantos argumentos como sea posible para agrupar las preferencias de los verbos, proponemos aquí un modelo general para aprender todas las preferencias correlacionadas en una oración, permitiéndonos medir la plausibilidad de su ocurrencia. Adicionalmente, este modelo nos permite usar tanto recursos estadísticos como recursos manuales como diccionarios o WordNet para mejorar la predicción. En particular, mostraremos un ejemplo del uso de PLSI, información mutua y WordNet para medir la plausibilidad.

5.1 Método

Primeramente construimos el recurso para contar las coocurrencias. Hacemos esto, como en los casos anteriores, analizando sintácticamente el corpus UKWaC con MINIPAR (Lin, 1988) para obtener una representación lematizada de dependencias. La oración *Poner una escena para amigos en el teatro* se convierte en

```
Poner obj:escena para:amigo en:teatro.
```

Después pre-calculamos las estadísticas de información mutua entre todos los pares de palabras, por ejemplo: (poner, obj:escena), (poner, para:amigo), (poner, en:teatro),

(obj:escena, para:amigo), (obj:escena, en:teatro), (para:amigo, en:teatro). Después procedemos a calcular la representación en tópicos para cada palabra usando PLSI.

5.2 Ensamblaje de las características para entrenamiento y prueba

Una vez que se construyen los recursos de PLSI y PMI, se analizan las oraciones de entrenamiento y prueba con MINIPAR, pero sólo se utiliza el primer nivel de análisis superficial. Asignamos las características a posiciones en un vector. Cada argumento tiene una posición fija, por ejemplo, el sujeto siempre irá en la primera posición, el objeto en la posición 75, los argumentos comenzando cono *en* en la posición 150, etc. De esta manera, las correlaciones pueden ser capturadas usando aprendizaje automático. En particular, usaremos una máquina de soporte vectorial (SVM). Hemos elegido un núcleo polinomial de segundo grado, de tal forma que pueda capturar las combinaciones de características. Cada una de las características de los argumentos se descompone en diversas subcaracterísticas. Estas subcaracterísticas consisten en la proyección de cada palabra en el espacio de tópicos de PLSI, la información puntual mutua (PMI) entre la palabra objetivo y la palabra característica, y la proyección de la palabra característica en el espacio de WordNet. La información puntual mutua se calcula como sigue:

$$PMI(t_1, t_2) = \frac{\log P(t_1, t_2)}{P(t_1, t_2)}$$

5.3 Experimentos

Como en los experimentos anteriores, realizaremos una tarea de pseudodesambiguación. Esta tarea consiste en cambiar una palabra objetivo (en este caso, el objeto directo) y después el sistema identificará la oración más plausible considerando el verbo y todos sus argumentos. Por ejemplo, para las oraciones 1) como arroz con palillos en la cafetería y 2) como bolsa con palillos en la cafetería, el sistema debería ser capaz de identificar la primera como la oración más plausible. Este experimento es similar a los previos mostrados en las secciones 2.1, 3.2 y 4.2, pero en este caso estamos considerando frases completas en lugar de sólo cuádruplas. Obtuvimos al azar 50 oraciones del corpus WSJ para los verbos: jugar, comer, sumar, calcular, fijar, leer, escribir, tener, aprender, inspeccionar, gustar, hacer, venir, ir, ver, parecer, dar, tomar, mantener, hacer, poner, enviar, decir, obtener, caminar, correr, estudiar, necesitar y devenir. Estos verbos fueron elegidos como una muestra de verbos altamente frecuentes, así como de verbos poco frecuentes. También son verbos que pueden tener una gran cantidad de argumentos, como *tomar*, es decir, su ambigüedad es alta. Para el entrenamiento, creamos conjuntos de palabras para los mismos verbos. Cada conjunto de entrenamiento contiene 125, 250 o 500 tripletas de dependencias para cada verbo. Cambiar el tamaño del entrenamiento nos permite contestar a la pregunta acerca de ¿qué tanta información se requiere por cada verbo para poder aprender algo significativo?

Los conjuntos de palabras se utilizaron tanto para entrenar al modelo PLSI como para crear la base de datos PMI. Después los mismos conjuntos de palabras se utilizaron para entrenar a la máquina de soporte vectorial (SVM). Cada oración fue tratada como una línea, tal como se describe en la sección 5.2, con cada característica expandida en subcaracterísticas de PLSI (tópicos). Generamos dos ejemplos falsos aleatoriamente por cada buen ejemplo, para que la SVM tenga ejemplos de cosas correctas, como de cosas incorrectas.

5.4 Adición de información manualmente obtenida

Como se describió en la sección 5.2, añadimos información manualmente obtenida a las tablas de entrenamiento y prueba. Esta información consiste en la distancia a los 38 conceptos superiores de WordNet, según fueron propuestos por Miller: tierra, objeto, ser, humano, animal, flora, artefacto, instrumento, dispositivo, producto, escritura, construcción, trabajador, creación, comida, bebida, locación, símbolo, sustancia, dinero, ropa, sentimiento, cambio de estado, movimiento, efecto, fenómeno, actividad, acto, estado, abstracción, atributo, relación, cognición, unidad, relación, tiempo y fluido.

Los resultados de los experimentos aparecen en la siguiente tabla.

Conjunto 125							
PMI	PLSI	WN	Aprendizaje	Cobertura	Precisión	Recup.	F
0	0	1	68.36%	89.44%	54.88%	49.09%	51.82%
0	1	0	89.59%	82.61%	66.96%	55.23%	60.53%
0	1	1	92.60%	**96.09%**	63.23%	60.76%	61.97%
1	0	0	93.63%	46.62%	**70.98%**	33.10%	45.15%
1	0	1	94.55%	94.88%	65.85%	**62.48%**	**64.12%**
1	1	0	97.14%	83.03%	66.09%	54.85%	59.95%
1	1	1	**98.01%**	**96.09%**	65.26%	62.71%	63.96%
Conjunto 250							
0	0	1	67.85%	89.49%	53.87%	48.21%	50.88%
0	1	0	88.01%	87.02%	69.44%	60.43%	64.62%
0	1	1	90.82%	**96.28%**	68.22%	**65.69%**	**66.93%**
1	0	0	93.24%	55.18%	**70.34%**	38.81%	50.02%
1	0	1	93.78%	95.39%	64.86%	61.87%	63.33%
1	1	0	96.88%	87.12%	68.99%	60.11%	64.24%
1	1	1	**97.28%**	**96.28%**	66.10%	64.64%	65.36%
Conjunto 500							
0	0	1	91.09%	89.49%	46.75%	41.84%	44.16%
0	1	0	86.75%	91.58%	68.32%	62.57%	65.32%
0	1	1	93.46%	96.79%	54.37%	52.63%	53.49%

1	0	0	92.95%	64.62%	65.11%	42.07%	51.11%
1	0	1	93.46%	95.72%	63.18%	60.48%	61.80%
1	1	0	96.65%	91.72%	**68.77%**	63.08%	**65.80%**
1	1	1	**96.68%**	**97.69%**	65.51%	**63.41%**	64.44%
Promedio							
0	0	1	91.09%	89.47%	51.83%	46.38%	48.96%
0	1	0	86.75%	87.07%	68.24%	59.41%	63.49%
0	1	1	93.46%	96.39%	61.94%	59.69%	60.80%
1	0	0	92.95%	55.47%	**68.81%**	37.99%	48.76%
1	0	1	93.46%	95.33%	64.63%	61.61%	63.08%
1	1	0	96.65%	87.29%	67.95%	59.35%	63.33%
1	1	1	**96.68%**	**96.69%**	65.62%	**63.59%**	**64.59%**

A partir de estos resultados, es posible ver que en la mayoría de los casos, combinar las tres fuentes de información mejora la tasa de aprendizaje, aunque separadamente, PMI provee la tasa de aprendizaje más alta. La cobertura siempre es mejor cuando se combinan los tres recursos, sin embargo, la precisión es mejor usando sólo PMI para pequeñas cantidades de datos de entrenamiento, en tanto que PLSI da mejor soporte cuando se añade más información. La recuperación es mayor para los casos que involucran la ayuda de información de WordNet. En promedio, a excepción de la precisión, los mejores valores se obtienen cuando se combinan los tres recursos.

6 Conclusiones y trabajo futuro

A pesar de la poca cantidad de datos de entrenamiento, hemos sido capaces de obtener tazas de predicción por encima de una línea base trivial de selección aleatoria entre dos opciones. Con estos experimentos fue posible determinar el impacto de usar diversos recursos, y además de medir el beneficio de usar un modelo en conjunto para aprendizaje con máquinas de soporte vectorial, en comparación con un simple modelo probabilístico de indexado semántico latente. Encontramos que al considerar todas las co-ocurrencias de los argumentos en una oración incrementa la recuperación en un 10%. También observamos que, como se esperaba, añadir más información incrementa la cobertura; sin embargo, la recuperación se incrementa en mayor medida usando máquinas de soporte vectorial sobre los modelos probabilísticos de indexado semántico, que usando éstos últimos únicamente.

Usar SVM incrementa la cobertura, la precisión y la recuperación, incluso cuando se entrena con la misma información disponible para PLSI. Esto sugiere que generar ejemplos negativos aleatoriamente, y aplicar aprendizaje automático a esta muestra, puede mejorar el desempeño de las tareas que utilizan modelos basados en tópicos.

Hemos propuesto un modelo que integra información estadística (PLSI y PMI) con recursos manualmente producidos como WordNet, y hemos probado que el desempeño se incrementa de esta manera, aunque el incremento no fue tan significativo como esperábamos. La mayoría de las características que contribuyen al desempeño vienen de PLSI. Sin embargo, el aprendizaje automático sobre PLSI tiene la ventaja de poder

capturar la correlación entre todos los argumentos, en oposición al modelo simple de PLSI.

Los trabajos futuros derivados de éste pueden considerar explorar un modelo matemático de tres variables basado en PLSI en lugar de una interpolación por pares, así como otras variaciones de iPLSI como uno basado en dos etapas, que consistiría en relacionar dos variables semánticas latentes con una variable latente en una segunda etapa.

Puesto que la prueba que realizamos produce alternativas aleatorias, nuestro sistema podría seleccionar candidatos más probables que el actual, por ejemplo, si en el texto existiera "vaca come heno en el patio", la alternativa generada automáticamente dijera "vaca come pasto en el patio", y el sistema seleccionara la segunda como más probable, sería considerada como un error, aunque podemos ver que no es así. Aunque se espera que el efecto de esto sea despreciable, debería ser considerado en futuros análisis.

Como trabajo futuro, planeamos evaluar con conjuntos de palabras más grandes, así como evaluar el desempeño de nuestro modelo en otras tareas como resolución de anáfora o detección de coherencia de oraciones.

Referencias

1. Agirre, E. and D. Martinez. 2001. Learning class-to-class selectional preferences, *Workshop on Computational Natural Language Learning*, ACL.
2. Baroni, M. and A. Lenci. 2009. One distributional memory, many semantic spaces. *Proceedings of the EACL 2009 Geometrical Models for Natural Language Semantics (GEMS) Workshop*, East Stroudsburg PA: ACL, 1–8.
3. Bergsma, S., D. Lin and R. Goebel, 2008. Discriminative Learning of Selectional Preference for Unlabeled Text. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 59–68
4. Bolshakov, I. A. 2005. An Experiment in Detection and Correction of Malapropisms through the Web, LNCS 3406, pp. 803-815.
5. Bolshakov, I.A., S. N. Galicia-Haro, A. Gelbukh. Detection and Correction of Malapropisms in Spanish by means of Internet Search. TSD-2005, Springer LNAI 3658: 115–122, 2005.
6. Budanitsky, E., and H. Graeme. Semantic distance in WorldNet: An experimental, application-oriented evaluation of five measures, NAACL Workshop on WordNet and other lexical resources, 2001.
7. Calvo, H., A. Gelbukh, and A. Kilgarriff. Automatic Thesaurus vs. WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment, Springer LNCS 3406:177–188, 2005.
8. Calvo, H., K. Inui and Y. Matsumoto. 2009. Interpolated PLSI for Learning Plausible Verb Arguments, In P*roceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pp.622–629.
9. Calvo, H., K. Inui, Y. Matsumoto. 2009a. Learning Co-Relations of Plausible Verb Arguments with a WSM and a Distributional Thesaurus. Procs. of the 14th Iberoamerican Congress on Pattern Recognition, CIARP 2009, Springer, Verlag. To appear.

10. Calvo, H., K. Inui, Y. Matsumoto. 2009b. Dependency Language Modeling using KNN and PLSI. Procs. of the 8th Mexican International Conference on Artificial Intelligence, MICAI 2009, Springer, Verlag, to appear.
11. Clarkson, P. R. and R. Rosenfeld. *Statistical Language Modeling Using the CMU-Cambridge Toolkit.* Procs. ESCA Eurospeech, 1997.
12. Deerwester, S., S. T. Dumais, G. W. Furnas, Thomas K. L, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, pp. 391–407.
13. Deschacht, K. and M. Moens. 2009. Semi-supervised Semantic Role Labeling using the Latent Words Language Model. Procs. 2009 Conf. on Empirical Methods in Natural Language Processing, *Proceedings of the 2009 conference on empirical methods in natural language processing (EMNLP 2009)*, pp. 21–29.
14. Ferraresi, A., E. Zanchetta, M. Baroni and S. Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. *Procs. of the WAC4 Workshop at LREC*. Marrakech, pp. 45–54.
15. Foley, W. A. Anthropological linguistics: An introduction. Blackwell Publishing, 1997.
16. Fuji A. and M. Iwayama (Eds.) Patent Retrieval Task (PATENT). Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, 2005.
17. Gao J., J. Y. Nie, G. Wu, and G. Cao, 2004. Dependence language model for information retrieval. Procs. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 170–177, 2004.
18. Gelbukh, A. and G. Sidorov, 1999. On Indirect Anaphora Resolution. PACLING-99, pp. 181-190, 1999.
19. Hoffmann, T. 1999. Probabilistic Latent Semantic Analysis, *Procs. Uncertainity in Artificial Intelligence'99, UAI*, 289–296.
20. Jiang J. and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the International Conference on Research in Computational Linguistics* ROCLING X.
21. Kawahara, D. and S. Kurohashi, 2001. Japanese Case Frame Construction by Coupling the Verb and its Closest Case Component, 1st Intl. Conf. on Human Language Technology Research, ACL..
22. Korhonen, Anna, 2000. Using Semantically Motivated Estimates to Help Subcategorization Acquisition. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong. 216-223.
23. Lee, L., 1999. Measures of Distributional Similarity, Procs. 37th ACL.
24. Lin, D. 1998a. Automatic Retrieval and Clustering of Similar Words. Procs. 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics.
25. Lin, D. 1998b. Dependency-based Evaluation of MINIPAR, Proc. Workshop on the Evaluation of Parsing Systems.
26. McCarthy, D. and J. Carroll. 2006. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Computational Linguistics* 29-4:639–654.
27. McCarthy, D., R. Koeling, J. Weeds, and J. Carroll, Finding predominant senses in untagged text. Procs 42nd meeting of the ACL, 280–287, 2004.

28. Merlo, P. and L. Van Der Plas. 2009. Abstraction and Generalisation in Semantic Role Labels: PropBank, VerbNet or both? *Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 288–296.
29. Ó Séaghdha, D. 2010. Latent variable models of selectional preference. *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics*, pp. 435–444.
30. Padó, S. and M. Lapata, 2007. Dependency-Based Construction of Semantic Space Models, Computational Linguistics 33-2: 161–199.
31. Padó, U. M. Crocker, and F. Keller, 2006. Modeling Semantic Role Plausibility in Human Sentence Processing, Procs. EACL.
32. Parton, K., K. R. McKeown, B. C., M. T. Diab, R. Grishman, D. Hakkani-Tür, M. Harper, H. Ji, W. Y. Ma, A. Meyers, S. Stolbach, A. Sun, G. Tur, W. Xu and S. Yaman. 2009. Who, What, When, Where, Why? Comparing Multiple Approaches to the Cross-Lingual 5W Task. 2009. Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 423–431.
33. Ponzetto, P. S. and M. Strube, 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution, Procs. Human Language Technology Conference, NAACL, 192–199.
34. Reisinger, J and Marius Paşca. 2009. Latent Variable Models of Concept-Attribute Attachment. *Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 620–628.
35. Resnik, P. 1996. Selectional Constraints: An Information-Theoretic Model and its Computational Realization, *Cognition*, 61:127–159.
36. Ritter, A., Mausam and Oren Etzioni. 2010. A Latent Dirichlet Allocation method for Selectional Preferences, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 424–434.
37. Rosenfeld, R., 2000. Two decades of statistical language modeling: where do we go from here?, Proceedings of the IEEE, Vol. 88, Issue 8, 2000, 1270–1278.
38. Salgueiro P., T. Alexandre, D. Marcu, and M. Volpe Nunes, 2006. Unsupervised Learning of Verb Argument Structures, Springer LNCS 3878, 2006.
39. Weeds, J. and D. Weir. 2003. A General Framework for Distributional Similarity, *Procs. conf on EMNLP*, Vol. 10:81-88.
40. Yamada I., K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. de Saeger, F. Bond and A. Sumida. 2009. Hypernym Discovery Based on Distributional Similarity and Hierarchical Structures. *Procs. 2009 Conf. on Empirical Methods in Natural Language Processing*, pp. 929–937.

Reviewing Committee

(Comité de revisión del volumen)

Ramón Zatarain Cabada, Instituto Tecnológico de Culiacán, Mexico
Carlos A. Reyes García, INAOE, Mexico
María Lucía Barrón Estrada, Instituto Tecnológico de Culiacán, Mexico
Yasmín Hernández Pérez, Instituto de Investigaciones Eléctricas, Mexico
Rafael Morales Gamboa, Universidad de Guadalajara, Mexico
Jaime Muñoz Arteaga, Universidad Autónoma de Aguascalientes, Mexico
Victor G. Sánchez Arias, CUAED UNAM, Mexico
Guillermo Rodríguez Ortíz, Instituto de Investigaciones Eléctricas, Mexico
Miguel Pérez Ramírez, Instituto de Investigaciones Eléctricas Mexico

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Septiembre de 2012
Printing 500 / Edición 500 ejemplares