Clasificación de servicios Web empleando estrategias supervisadas

Víctor O. Díaz-Torres, Esaú Villatoro-Tello,
Christian Sánchez-Sánchez y Héctor Jiménez-Salazar

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana Unidad Cuajimalpa, México DF
{209363180, evillatoro, csanchez, hjimenez}@correo.cua.uam.mx

Resumen El incremento de Servicios Web disponibles en catálogos públicos en Internet ha provocado la urgente necesidad de proponer nuevas formas de búsqueda y categorización de los mismos. El lenguaje de descripción WSDL permite a los proveedores de Servicios Web dar detalles del funcionamiento de un servicio al momento de registrarlo en algún catálogo. El problema principal de esta técnica es que se depende en gran medida de la experiencia y criterio del proveedor al momento de asignar una categoría. Dentro de este trabajo se propone una nueva forma de representación de los Servicios Web que aprovecha la información contenida en los atributos de descripción contenidos en el WSDL. Nuestra propuesta emplea técnicas tradicionales de clasificación de textos basadas en aprendizaje supervisado. Los resultados obtenidos muestran que es posible obtener resultados eficientes en la clasificación aún sin contar con el atributo *Documentation* de los Servicios Web.

Palabras clave: servicios web, documentos WSDL, clasificación de textos, aprendizaje supervisado, procesamiento de lenguaje natural.

1. Introducción

Actualmente, en el ámbito del desarrollo de Software existe una fuerte motivación para preservar y practicar las *buenas costumbres de programación*. Entre las ventajas que estas buenas costumbres de programación ofrecen están: *i)* alto rendimiento de aplicaciones al permitir su funcionamiento de manera distribuida, *ii)* colaboración a través de mecanismos estandarizados, *iii)* reutilización (reuso) de software, y, *iv)* reducción de costos en los procesos de desarrollo de software.

Los Servicios Web (WS, por sus siglas en Inglés) son un ejemplo de los mecanismos que surgieron a partir del conjunto de necesidades mencionadas en el párrafo anterior. Agregado a esto, los WS surgieron como una tendencia de negocios sobre aplicaciones, que en el fondo contienen descripciones encapsuladas de una aplicación (métodos y/o funciones) que son de uso público y que están

basados en estándares de Internet, tales como lo son los lenguajes WSDL[1], OWL-S o WSMO.

En años recientes el número WS ha crecido considerablemente, lo cual dificulta la tarea de buscar y localizar efectivamente un WS. Agregado a esto, para la mayoría de las personas no es fácil leer e interpretar la información (descripción y mensajes para su ejecución) referente a los WS. Con la finalidad de apoyar en la solución a este problema han surgido registros públicos en los cuales se propone un estándar a seguir cada vez que se produce un nuevo WS. El problema con este tipo de servicios es que dependen en gran manera del criterio y experiencia del proveedor al momento de registrar su WS.

Con la finalidad de poder dar soluciones efectivas a los problemas antes mencionados, en este trabajo se propone una técnica de clasificación automática de WS que aprovecha los componentes de la descripción WSDL de un servicio Web. WSDL (Web Service Description Language) es una de las formas más comúnmente empleadas para definir los servicios web. Una descripción en WSDL aprovecha las ventajas ofrecidas por la gramática de XML para definir los medios de comunicación, el intercambio de mensajes y las reglas de operación de un WS. Dentro de un documento WSDL, la definición abstracta de las operaciones y de los mensajes se encuentran separadas de su implementación concreta, lo que permite la reutilización de las definiciones abstractas: mensajes, los cuales son descripciones abstractas de los datos que son intercambiados, y de los tipos de puertos los cuales son colecciones abstractas de las operaciones.

Nuestros experimentos muestran que a través de utilizar ciertos componentes de la descripción WSDL es posible alcanzar resultados prometedores al momento de realizar la clasificación de servicios Web. Nuestra propuesta emplea técnicas tradicionales de Procesamiento de Lenguaje Natural para representar los documentos WSDL y además, por medio de una estrategia de aprendizaje supervisado se construye un modelo de clasificación de WS.

El resto del documento se encuentra organizado de la siguiente manera. En la sección 2 se describe brevemente el trabajo relacionado más relevante a la temática en cuestión. En la sección 3 se describe el método propuesto así como las técnicas empleadas. La sección 4 detalla los conjuntos de datos, la configuración experimental y los resultados obtenidos. Finalmente, la sección 5 muestra las conclusiones obtenidas y define las líneas de trabajo futuro.

2. Trabajo relacionado

En [3] se propone un enfoque para clasificar y anotar semánticamente Servicios Web. La clasificación de los servicios a un dominio especifico lo realizan usando Maquinas de Soporte Vectorial (MSV), posteriormente identifican conceptos clave en la documentación de los servicios, los cuales son relacionados por medio de un enrejado de conceptos para anotar los servicios. Los autores reali-

[1] Algunos ejemplos catálogos de WS disponibles a través de UDDIs (Universal Description, Discovery and Integration) en Internet son Xmethods [1] y Seekda [2].

zaron pruebas con un conjunto de 205 Servicios Web distribuidos en 11 clases[2], y obteniendo una precisión en la clasificación de 63 %.

Un trabajo similar es fue el realizado por [4], donde combinaron la clasificación de documentos y la alineación de ontologías para enriquecer semánticamente las descripciones de los Servicios Web. Los autores clasifican los servicios a partir de calcular el centroide de cada categoría, para que posteriormente cuando llegue un nuevo documento este sea incluido a la categoría que tenga el centroide más cercano. Posteriormente, relacionan la información en la descripción del servicio con conceptos en una ontología y ya teniendo los servicios anotados, entonces proceden a buscar conceptos similares en otras ontologías. Las pruebas las realizaron con un conjunto de 391 Servicios Web[3] tomando un subconjunto de 100 descripciones de servicio que anotaron semánticamente de manera manual obteniendo un total de 6 categorías. Los resultados que ellos reportan son con respecto a la precisión del anotado, de acuerdo a una métrica que proponen, donde obtuvieron un promedio de la precisión de la anotación del 85.7 %, ya que reportan que el anotado semántico ayuda a corregir los errores del modelo de clasificación.

Otra forma de clasificar Servicios Web, fue la propuesta en [5], en la cual a partir de servicios anotados semánticamente proponen una serie de heurísticas para determinar a qué categoría pertenecen los Servicios Web, ellos proponen comparar el servicio con todos los servicios de cada categoría verificando a cuales es más parecido, en la comparación verifican la similitud de las conceptos (en ontologías) relacionados con las operaciones del servicio y de sus entradas y salidas. Los autores utilizaron un corpus de 164 descripciones de Servicios Web (anotados semánticamente) pertenecientes a 23 diferentes categorías[4]. Ellos obtuvieron un promedio de éxito en la clasificación de 83 %, también debida a la anotación semántica.

En [6] clasifican por medio de máquinas de soporte vectorial, los Servicios Web de acuerdo a la categoría UNSPSC (United Nations Standard Products and Services Code). En este trabajo ellos toman las características funcionales del servicio (nombre de operaciones, entradas y salidas). Para hacer la clasificación, y debido a que UNSPC es una jerarquía de categorías, ellos emplean primero clasificadores que distinguen las categorías padre y posteriormente subclasificadores que ayudan a determinar la categoría dado ese padre. También obtienen términos similares por medio de WordNet, ya que las MSV pueden trabajar con vectores con muchos términos. Los autores trabajaron con una colección de 1007 Servicios Web anotados semánticamente[5], preclasificados en siete categorías. Hicieron dos tipos de experimentos: en el primero utilizaron tres diferentes medidas de similitud semántica en dos diferentes funciones *kernel* de MSV, el mejor resultado lo obtuvieron con la medida de similitud Path Length y RBF de función kernel con el 90.5 % de exactitud. Y en los segundos experi-

[2] Disponibles en *http://moguntia.ucd.ie/repository/ws2003*
[3] Disponibles en *http://www.andreas-hess.info/projects/annotator/*
[4] Disponibles en *http://www.andreas-hess.info/projects/annotator/*
[5] Descargados de *http://projects.semwebcentral.org/projects/owls-tc/*

mentos solo seleccionando 500 Servicios Web utilizando la medida de similitud y el kernel que les dio mejores resultados, compararon con otros métodos de clasificación (MSV con información semántica y textual como características de clasificación y método de comparación propiedad-valor) su método gano con el 90.5 % mostrado previamente.

El enfoque propuesto dentro de este trabajo se diferencia del trabajo previo principalmente en que no utiliza fuentes externas de conocimiento para llevar a cabo la representación de las descripciones de los servicios Web. Agregado a esto se propone un modelo más realista, pues los Servicios Web que se encuentran en Internet son muy heterogéneos y no siempre cuentan con toda la información de sus componentes. Por ello nos orientamos a seleccionar combinaciones de características extraídas de los Servicios Web para validar cuales de estás nos ayudan a una mejor y menos costosa clasificación.

3. Método propuesto

3.1. Pre-procesamiento de los documentos WSDL

El pre-procesamiento consiste en la extracción de las palabras contenidas en los atributos de la descripción de los WS. Los atributos de descripción considerados para nuestros experimentos fueron: *nombre del servicio, documentación, nombre de los mensajes y nombre de los parámetros*. Es importante mencionar que de todos estos atributos, el único que se encuentra en lenguaje natural es el de *documentación*, por lo tanto su pre-procesamiento es relativamente sencillo. Por el contrario, para el resto de los atributos (*i.e., nombre del servicio, mensajes y parámetros*) fue necesario definir un conjunto de heurísticas más elaboradas para poder extraer elementos relevantes para nuestros fines. Para la realización de nuestros experimentos se aplicaron el mismo conjunto de reglas definidas en [8], las cuales son:

- Filtrado de términos. Utilizando un diccionario, definido a través de *Wordnet*, se filtran palabras desconocidas de los atributos. El objetivo es conservar sólo términos conocidos para la representación de los WS.
- División de términos compuestos. Para esto se emplean las técnicas definidas en [8], donde el objetivo es descubrir términos adicionales que posiblemente pueden aportar información valiosa a la representación del WS.
- Eliminación de palabras vacías. Las palabras vacías representan palabras carentes de información tales como: artículos, preposiciones, pronombres, etc. Al eliminar este tipo de palabras es posible reducir la cantidad de *ruido* en la representación de los WS.

3.2. Representación de los WS

Como se ha venido mencionando en secciones previas, atacamos el problema de la clasificación de servicios web empleando el paradigma de clasificación de

textos (CT)[6]. Bajo este paradigma un primer paso necesario es el *indexado* de los documentos de entrenamiento (Tr), actividad que corresponde al mapeo de un documento d_j en una forma compacta de su contenido. La representación más comúnmente utilizada para representar cada documento es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información [7]. Es decir, un texto d_j es representado como el vector $\overrightarrow{d}_j = \langle w_{kj}, \ldots, w_{|\tau|j} \rangle$, donde τ es el *diccionario*, *i.e.*, el conjunto de términos que ocurren al menos una vez en algún documento de Tr, mientras que w_{kj} representa la importancia del término t_k dentro del contenido del documento d_j. En ocasiones τ es el resultado de filtrar las palabras del vocabulario, *i.e.,* resultado de un preprocesamiento (sección 3.1). Una vez que hemos hecho los filtrados necesarios, el diccionario τ puede definirse de acuerdo a diferentes criterios, sin embargo el que se empleó en esta propuesta corresponde a la Bolsa de Palabras.

La Bolsa de Palabras (BOW)[7] es la forma tradicionalmente utilizada para representar los documentos [9]. Este método de representación utiliza a las palabras simples como los elementos del vector de términos.

Con respecto al peso (*i.e.,* la importancia) w_{kj}, se tienen diferentes formas de calcularlo, entre las más usadas en la comunidad científica se tienen el ponderado booleano, ponderado por frecuencia de término y el ponderado por frecuencia relativa de términos. Una breve descripción es dada a continuación:

- *Ponderado Booleano:* Consiste en asignar el peso de 1 si la palabra ocurre en el documento y 0 en otro caso.

$$w_{kj} = \begin{cases} 1, \text{si } t_k \in d_j \\ 0, \text{en otro caso} \end{cases} \tag{1}$$

- *Ponderado por frecuencia de termino (TF):* En este caso el valor asignado es el número de veces que el término t_k ocurre en el documento d_j.

$$w_{kj} = f_{kj} \tag{2}$$

- *Ponderado por frecuencia relativa (TF-IDF):* Este tipo de ponderado es una variación del tipo anterior y se calcula de la siguiente forma:

$$w_{kj} = TF(t_k) \times IDF(t_k) \tag{3}$$

  donde $TF(t_k) = f_{kj}$, es decir, la frecuencia del termino t_k en el documento d_j. IDF es conocido como la "frecuencia inversa" del termino t_k dentro del documento d_j. El valor de IDF es una manera de medir la "rareza" del termino t_k. Para calcular el valor de IDF se utiliza la siguiente formula:

$$IDF(t_k) = \log \frac{|D|}{\{d_j \in D : t_k \in d_j\}} \tag{4}$$

  donde D es la colección de documentos que está siendo indexada.

[6] La Clasificación de Textos es la tarea de asociar automáticamente categorías predefinidas con documentos a partir del análisis de su contenido [9].

[7] En inglés se dice "Bag of Words".

3.3. Algoritmo de aprendizaje

En la literatura referente a la CT [9] existe gran variedad de algoritmos que han sido evaluados y que han mostrado ser apropiados para la tarea de clasificación de textos. Dado que el objetivo de este trabajo no es evaluar exhaustivamente cuál podría ser el mejor clasificador o aprendiz, si no más bien evaluar la pertinencia de la representación propuesta; mostramos resultados solo al trabajar con el algoritmo de *k-means*[8] [10].

El algoritmo *k-means* calcula en un primer paso un valor *media* o *centroide* m_i para cada una de las clases. Cuando se proporciona un nuevo elemento sin clasificar, este es asignado a la clase i con cuyo centroide m_i se tenga la menor distancia, *i.e.*, la mayor similitud. Para nuestros experimentos, el centroide de cada una de las categorías de WS es definido siguiendo una forma de representación vectorial como se explico en la sección 3.2.

Para el calculo de similitud se han propuesto varias métricas que permiten determinar el parecido de pares de documentos [11]. El objetivo de estas métricas es contar con un valor numérico al cual llamaremos coeficiente de similitud SC, el cual nos dirá cuán parecidos son los documentos D_i y D_j, note que cualquiera de estos dos documentos puede representar el centroide m_i. Dos medidas ampliamente utilizadas en el campo de recuperación de información que permiten determinar la similitud entre documentos son:

- **Medida Cosenoidal.** La idea básica de ésta es medir el ángulo entre el vector de D_i y de D_j, para hacerlo, calculamos:

$$SC(D_i, D_j) = \frac{\sum_{k=1}^{t} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{t} (w_{jk})^2 \sum_{k=1}^{t} (w_{ik})^2}} \tag{5}$$

- **Medida DICE.** El coeficiente de DICE es obtenido por medio de:

$$SC(D_i, D_j) = \frac{2 \sum_{k=1}^{t} w_{ik} w_{jk}}{\sum_{k=1}^{t} (w_{jk})^2 + \sum_{k=1}^{t} (w_{ik})^2} \tag{6}$$

En todos los casos k va de 1 a el número total de términos del vocabulario τ, w_{ik} indica la importancia del término k en el documento D_i mientras que w_{jk} la importancia del término k en el documento D_j.

3.4. Evaluación

Para evaluar un sistema de clasificación de textos se utilizan las medidas de *Precisión* y *Recuerdo*, que son medidas comunes en el área de recuperación de información. La precisión (P) es la proporción de documentos clasificados correctamente en una clase c_i con respecto a la cantidad de documentos clasificados en esa misma clase. El recuerdo (R), la proporción de documentos clasificados

[8] El algoritmo *k-means* es también conocido como *k-medias*

correctamente en una clase c_i con respecto a la cantidad de documentos que realmente pertenecen a esa clase. Así, la precisión se puede ver como una medida de la corrección del sistema, mientras que el recuerdo da una medida de cobertura o completitud.

Normalmente se emplea la medida F para describir el comportamiento de la clasificación, la cual se define como:

$$F = \frac{(1+\beta^2)Precision * Recuerdo}{\beta^2 Precision + Recuerdo} \tag{7}$$

donde con $\beta = 1$ representa la media armónica entre la precisión y el recuerdo. La función de β es la de controlar la importancia relativa entre las medidas de precisión y recuerdo. Es común asignar un valor de 1 indicando igual importancia a ambas medidas.

4. Experimentos y resultados

4.1. Conjunto de datos

Para nuestros experimento se trabajo con la colección de ASSAM[9]. La colección fue filtrada aplicaron las mimas técnicas que se describen en [8], resultando en un total de 203 documentos WSDL repartidos en 22 clases.

4.2. Definición experimentos

Un conjunto de 7 configuraciones de experimentos fueron propuestos, los cuales consistieron en determinar el aporte de información de cada uno de los atributos extraídos de los documentos WSDL. Como se mencionó en la sección 3.1, los atributos de descripción considerados fueron: *documentación (Doc), nombre servicio (Nom), mensajes (Msgs)* y *parámetros (Param)*. La configuración base de nuestros experimentos esta dada por el uso únicamente del atributo *Doc*; idealmente todo proveedor de un WS deberá proporcionar una documentación lo suficientemente clara que permita hacer una clasificación correcta del mismo.

Agregado a esto se consideró las combinaciones de atributos siguientes: *Nom + Msgs, Nom+Param* y *Nom+Msgs+Param*, donde la finalidad de estos experimentos fue demostrar que es posible determinar la categoría de un WS sin la necesidad de contar con una *documentación*. La tabla 1 muestra el tamaño del vector del vocabulario para cada uno de los experimentos propuestos.

4.3. Resultados

La gráfica mostrada en la figura 1 muestra los resultados obtenidos por el clasificador *k-means* al emplear como medida de similitud el coeficiente DICE.

[9] http://www.andreas-hess.info/projects/annotator/

Tabla 1. Tamaño del vector de características correspondiente a cada uno de los atributos de descripción considerados en los experimentos.

Experimento	Tam. Vocabulario
Doc	2599
Nomb	748
Param	799
Msgs	656
Nomb+Msgs	755
Nomb+Param	930
Nomb+Msgs+Param	935

Las gráficas muestran el desempeño del clasificador en términos de la medida F (sección 3.4).

Nótese que los resultados obtenidos muestran que el uso de *Doc* resulta en un desempeño bajo del clasificador, apenas logrando un 0.72 bajo un esquema de pesado TF-IDF. De igual forma los resultados muestran que manejar sólo *Nom*, *Msgs* y/o *Param* no permite mejorar los resultados notoriamente, sin embargo, es importante mencionar que el tamaño del vector de estas configuraciones es de menos de la mitad comparado con el empleado por *Doc* (tabla 1), lo cual indica que contienen información relevante para el clasificador.

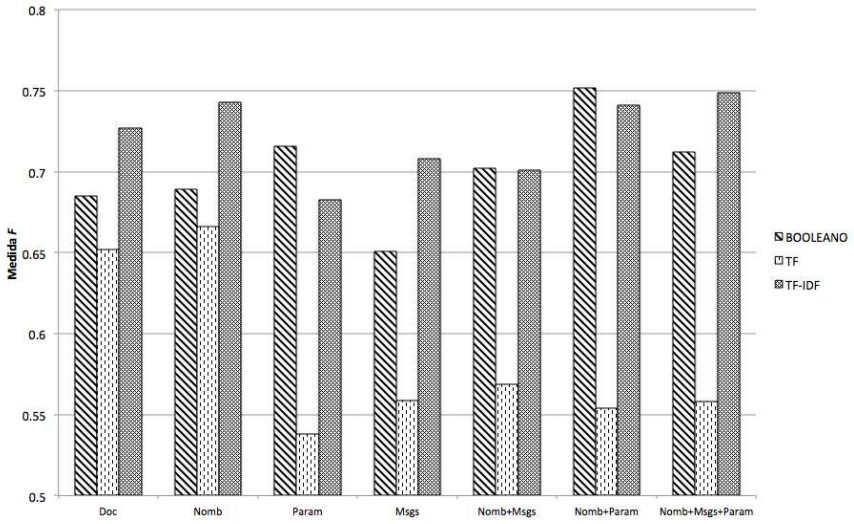

Figura 1. Resultados obtenidos empleando como medida de similitud el coeficiente DICE.

Los mejores resultados que se obtuvieron fue bajo la configuración de *Nom* + *Param* bajo un esquema de pesado *booleano*, alcanzando un desempeño de 0.75 en la medida F. Otro factor importante que vale la pena resaltar es el hecho

de que manejar un esquema de *presencia* de los atributos, *i.e.,* un esquema de pesado *booleano*, aporta mayor información al clasificador que los esquemas de *frecuencia* y *frecuencia relativa* respectivamente.

La gráfica mostrada en la figura 2 muestra los resultados obtenidos por el clasificador *k-means* al emplear como medida de similitud el Coseno.

Como se puede observar, la configuración *Nom+Param* supera de manera considerable al método base logrando un 0.81 en la medida F contra un 0.72 obtenido al usar sólo *Doc*. Nóte también, que el resultado de estos experimentos muestra el mismo comportamiento de los mostrados en la tabla 1, *i.e.,* una forma de representación *booleana* aporta mejores elementos al clasificador que las frecuencias.

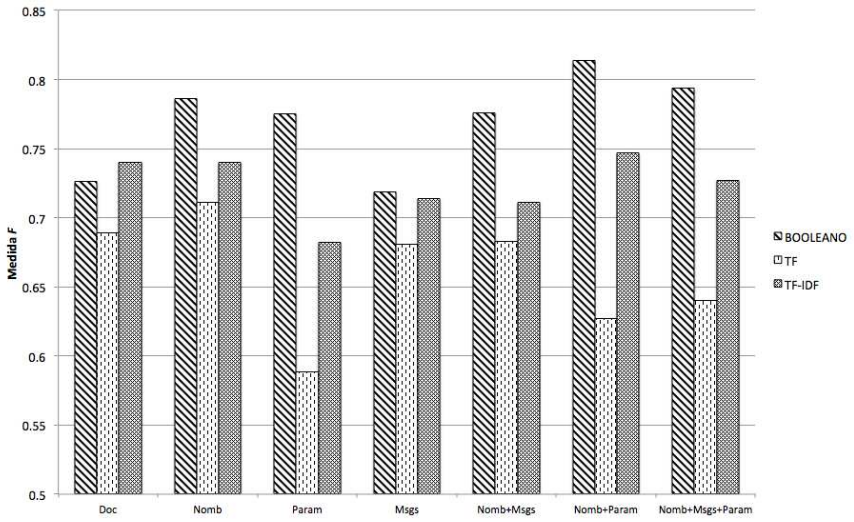

Figura 2. Resultados obtenidos empleando como medida de similitud el Coseno.

Finalmente, los resultados mostrados en la figura 2 muestran que el uso de una medida de similitud más fina (*e.g.,* el Coseno) permite distinguir mejor entre las categorías de los servicios Web.

5. Conclusiones

En este trabajo se ha propuesto una nueva forma de representación y de clasificación de Servicios Web que aprovecha las ventajas de la información contenida en las descripciones WSDL. El método propuesto emplea técnicas de Clasificación de Textos que han sido tradicionalmente empeladas en el área de Procesamiento de Lenguaje Natural.

Se mostró un estudio del aporte que tienen los atributos de descripción contenidos en un WSDL. Los resultados obtenidos muestran que es posible prescindir

de la existencia de la *documentación* y aún así obtener buenos resultados durante la clasificación. Los resultados obtenidos son alentadores, pues nos permiten afrontar la problemática de las ausencia de *documentación* en los documentos WSDL, el cual es un problema muy frecuente.

Cómo trabajo futuro se pretende evaluar el aporte que podrían tener representaciones basadas en n-gramas[12]. Los n-gramas han mostrado ser efectivos en tareas de clasificación de textos gracias a que permiten capturar información del contexto agregado a que permiten mantener el orden de aparición de las palabras, cosa que no sucede con una representación de BOW. Intuitivamente, el uso de n-gramas permitirá tener una representación más fina con la cual podrían mejorarse los resultados de clasificación.

Agradecimientos. Agradecemos a la Universidad Autonoma Metrópolitana Unidad Cuajimalpa y al proyecto CONACYT número CB2010/153315 por el apoyo para la asistencia a este evento.

Referencias

1. Xmethods, *http://xmethods.net/ve2/index.po* (Ultima visita en Abril de 2013)
2. Seekda, *http://webservices.seekda.com/* (Ultima visita en Noviembre 2012)
3. Bruno, M., Canfora, G., Di Penta, M., Scognamiglio, R. (2005) An Approach to support Web Service Classification and Annotation. En *Proceeding EEE '05 Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05).* IEEE Computer Society Washington.
4. Crasso, M., Zunino, A., Campo, M. (2010) Combining Document Classification and Ontology Alignment for Semantically Enriching Web. En *New generation computing.* Vol. 28, No. 4, pp. 371-403.
5. Corella, M. A. y Castells, P. (2006) Semi-automatic semantic-based web service classification. En *Proc. of the International Conference on Knowledge-Based Intelligent Information and Engineering Systems.*
6. Wang, H., Shi, Y., Zhou, X., Zhou, Q., Shao, S. y Bouguettaya, A. (2010) Web Service Classification Using Support Vector Machine. En *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI).* pp.3-6
7. Baeza-Yates, R., y Ribeiro-Neto, B. (1999) Modern Information Retrival, Addison Wesley.
8. Jiménez-Salazar, H., Sánchez-Sánchez, C., Rodríguez-Lucatero, C. y Luna-Ramírez, A. W. (2012) An Analysis of Web Services Attributes for Discovery Support. En *Research In Computing Science*
9. Sebastiani F. (2002) Machine Learning in Automated Text Categorization. En *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1–47.
10. Duda, O., Hart, P.E., y Stork, D.G. (2000) En Pattern Classification, John Wiley & Sons.
11. Grossman, D. A. y Frieder, O. (2004) En *Information Retrieval*, Algorithms and Heuristics. Springer, second edition edition.
12. Sidorov G., Velasquez F., Stamatatos E., Gelbukh A., Chanona-Hernández L..(2013) Syntactic Dependency-Based N-grams: More Evidence of Usefulness in Classification. CICLing 2013. En *Lecture Notes in Computer Science.* Vol. 7816, pp. 13-24.