

A Document Recommendation System Using a Document-Similarity Ontology

R. Vences, J. Gómez and V. Menéndez

Abstract— In this paper, we present a proposal for searching, retrieving and recommending documents, based on the similarity of their content, using an ontology of the type “Friend of a Friend” that establishes, if it is the case, the transitivity of similarity between the document selected by the user and the other documents in the repository. Document similarity is calculated as the cosine of the angle between the vectors of terms that represent them. The Vector Space Model is used for representing the documents and the weight of the terms is obtained by using the TF-IDF metric. The architecture that implements the proposal is presented as well as experimental results.

Keywords— Recommender Systems, FOAF Ontology, TF-IDF, Cosine Similarity.

I. INTRODUCCIÓN

EL CRECIMIENTO exponencial que ha tenido Internet [1] en la última década junto con la creciente disponibilidad de documentos en formato digital, almacenados muchas veces de manera no estructurada, han planteado un gran desafío científico y tecnológico en muchos aspectos, en particular para manipular, buscar y recuperar la información y el conocimiento contenido [2].

Existen herramientas para manejar el gran volumen de información como por ejemplo: clasificadores temáticos de textos, sistemas de detección (de plagio, de reputación en redes sociales). Sin embargo, todos tienen en común conocer el grado de similitud textual entre un par de textos [3].

Surge entonces la necesidad de procesar automáticamente ese gran volumen de información y para ello se ha recurrido a una de las ramas de la Inteligencia Artificial: el Procesamiento de Lenguaje Natural (PLN) [4]. El enfoque principal de esta área consiste en crear métodos, técnicas y herramientas computacionales que permitan realizar análisis de información escrita u oral y que faciliten la búsqueda y organización de dicha información [5].

Dentro de los esfuerzos que se han realizado para gestionar la información textual se encuentran principalmente los del área de la recuperación de información (RI), la cual trata con la representación, almacenamiento, organización y acceso de elementos de información. Los documentos recuperados deben satisfacer las necesidades de información del usuario expresadas en lenguaje natural [6].

La RI significa encontrar material (usualmente documentos) de naturaleza no estructurada (generalmente texto) que satisface una necesidad de información desde

dentro de grandes colecciones (comúnmente almacenados en computadoras) [7].

El uso del PLN juega un papel primordial en la mejora de la eficacia en el uso de herramientas disponibles para el acceso y la búsqueda de información en la Web (por ejemplo usando sinónimos de palabras contenidas en nuestra petición de búsqueda), aprovechando así sus aspectos semánticos [8].

Cuando utilizamos la Web como plataforma para la búsqueda y recuperación de información, un elemento muy importante son las ontologías, ya que permiten definir conceptos básicos y sus relaciones representando un dominio de conocimiento, en un formato estándar, alcanzable y manejable por las tecnologías de la Web Semántica (RDF, OWL, SKOS, SPARQL, etc.) [9].

Una ontología es una especificación explícita y formal de una conceptualización compartida [10]. En una ontología se definen entidades, clases, propiedades predicados y relaciones entre estos componentes, modelando un dominio de conocimiento.

Por otra parte, los repositorios suelen almacenar en bases de datos gran cantidad de información sobre los documentos, los usuarios y sus interacciones. Los repositorios actúan como filtros para los recursos referenciados y almacenados, pues la única forma de acceder a ellos es por medio de una búsqueda basada en los metadatos almacenados [11].

El objetivo de este trabajo es ofrecer una solución para la búsqueda, recuperación y recomendación de documentos con contenido similar a aquellos seleccionados por el usuario, utilizando una ontología que modele los documentos y las relaciones de similitud entre los mismos.

La similitud entre los documentos se calcula con base en su contenido, utilizando la frecuencia de los términos (métrica tf-idf) y el coseno del ángulo entre los vectores que los representan. Para la representación de los documentos se utiliza el modelo espacio vectorial [12].

El contenido del artículo está organizado de la siguiente manera. En la sección I se presentó una breve introducción a la problemática detectada. En la sección II se presenta la arquitectura del sistema, explicando cada uno de sus componentes, sus relaciones y el flujo de datos entre ellos. La sección III detalla la metodología utilizada, mientras que la sección IV presenta la implementación. En la sección V se describe la experimentación y se presenta el análisis de los resultados obtenidos. Finalmente la sección VI aborda las conclusiones y las sugerencias para trabajos futuros.

II. ARQUITECTURA DEL SISTEMA

En la Fig. 1 se muestra la arquitectura del sistema, la cual propone una solución para la búsqueda, recuperación y recomendación de documentos. El usuario ingresa al sistema

R. Vences, Universidad Autónoma de Yucatán, vnava@correo.uady.mx.
 J. Gómez, Universidad Autónoma de Yucatán, jgomez@correo.uady.mx
 V. Menéndez, Universidad Autónoma de Yucatán, mdoming@correo.uady.mx

una consulta (palabras clave), las cuales son sometidas al mismo pre-procesamiento que los documentos. Esta consulta es la entrada del módulo de *búsqueda y recuperación de información*, que devuelve un subconjunto de documentos del repositorio ordenados por relevancia de acuerdo a la función *match* de mysql, que se basa en la frecuencia de términos con respecto a la consulta [13].



Figura 1. Arquitectura del sistema.

Cuando el usuario selecciona algún documento, el sistema ejecuta el módulo *similitud de artículos*, que hace uso de la ontología para obtener los documentos relacionados por medio de la propiedad *relacionadoCon* del documento seleccionado.

El sistema identifica al usuario por medio de su dirección IP y solicita al módulo de *perfil de usuario*, la información del mismo, como por ejemplo, los documentos que anteriormente ha consultado, las palabras clave que ha utilizado, las valoraciones que ha realizado a los documentos, etc.

Con la información del perfil del usuario y los trabajos relacionados, el módulo de *recomendación de artículos* hace un filtrado de la información presentándole al usuario la lista de documentos que estén relacionado con el documento que seleccionó, tanto los que ya ha consultado, como los que se le recomienda que consulte.

El sistema también mostrará los documentos relacionados en segundo grado, esto es, aquellos que estén relacionados por medio de la propiedad *relacionadoCon* con los documentos relacionados directamente del documento seleccionado.

El módulo de pre procesamiento e indexación de información se realiza fuera de línea y es el encargado de realizar el cálculo de similitudes entre los documentos y alimentar con esos datos a la ontología.

En la Fig. 2 se muestra un diagrama de flujo del sistema en línea.

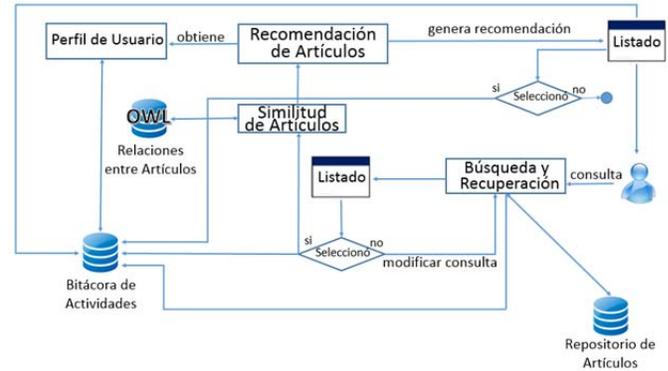


Figura 2. Diagrama funcional del sistema en línea.

III. METODOLOGÍA

Es preciso contar con una forma consistente de representar cada documento (su contenido), para ello se utilizará el modelo espacio vectorial [12], que es uno de los más empleados. El modelo vectorial básicamente consiste en que cada documento puede ser representado mediante un vector de términos. Cada término lleva asociado un coeficiente o peso que trata de expresar la importancia o grado de representatividad de ése término en ese vector o documento. Este peso puede calcularse de forma automática a partir de diversos elementos: basándose en la frecuencia de términos, tanto en toda la colección de documentos con que se trabaje, como dentro de cada documento en particular.

La creación del sistema de pesos basada en *Term Frequency – Inverse Document Frequency (TF-IDF)* se fundó en la teoría de Spärk-Jones que indicaba que los términos con altas frecuencias pueden ser útiles para aumentar la cobertura, pero las correspondencias entre las consultas y los términos del documento que ocurren raramente en una colección de documentos deberían de ser tomadas como más importantes que aquellas que ocurren frecuentemente [14].

Para construir el vector de términos de un documento es necesario aplicar un procesamiento a los documentos para extraer la información textual. Seguidamente se elimina del texto los caracteres que no son relevantes, como por ejemplo acentos, números, tabuladores, etc. Por último se eliminan las palabras vacías (*stop words*), utilizando un diccionario del Inglés, debido a que los documentos están en ese idioma.

La eliminación de palabras vacías es el proceso mediante el cual se excluyen del proceso de representación aquellas palabras de frecuente aparición que no aportan ningún significado a un documento, sino que se utilizan para seguir las reglas del idioma. Este es el caso de las preposiciones, conjunciones, determinantes, artículos, pronombres, etc.

Dado que la eliminación de palabras vacías es una tarea clásica dentro del pre procesamiento lingüístico de cualquier texto, existen diversas listas tanto en la literatura [15] como en Internet.

La tarea siguiente es la construcción de un índice invertido que relacione términos con documentos en los cuáles ese término aparece y en cuántas ocasiones lo hace, con el fin de tener la información suficiente para obtener su representación vectorial.

Para este momento ya se tiene por cada documento original un conjunto de palabras (términos) los cuales servirán

para obtener su representación vectorial (vector de términos) utilizando la ecuación (1) para calcular la relevancia de cada término.

$$tf-idf_i = tf_{ij}/M * \log(N/n_i) \tag{1}$$

donde, tf_{ij} es la frecuencia del término i en el documento j , M es la norma del vector correspondiente al documento j , N es el número total de documentos y n_i es el número de documentos donde aparece el término i .

Esta ecuación combina la frecuencia del término en el documento con la frecuencia del término dentro de la colección con el objetivo de reflejar el poder de discriminación del término el cual vendrá dado por su aparición con frecuencias altas en varios documentos individuales, junto con su escasa ocurrencia en la colección completa. De esta manera, se determinará que la importancia del término dentro del documento crece con respecto a la frecuencia del término dentro de él y disminuye con respecto al número de documentos que lo contienen [15].

La similitud coseno como se puede observar en la ecuación (2), es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos [16]. Si los documentos son iguales, el ángulo vale 0 y el coseno 1. En cambio, si son ortogonales el coseno vale 0.

$$\text{Cos}(d_1, d_2) = (d_1 \cdot d_2) / (|d_1| \times |d_2|) \tag{2}$$

donde, $d_1 \cdot d_2$ es el producto punto entre ambos vectores y $(|d_1| \times |d_2|)$ es el producto de la norma de ambos vectores.

Una vez obtenida la representación vectorial de los documentos y haber calculado la similitud de cada documento contra el resto de la colección, se almacena esa información en una base de datos que después servirá para crear las instancias que reflejan las relaciones de la transitividad de la similitud entre documentos.

Para la construcción de la ontología se utilizó Protégé [17], siguiendo algunas reglas y sugerencias, aunque como menciona Noy and McGuinness, “no hay una sola ontología correcta para un dominio dado” [18].

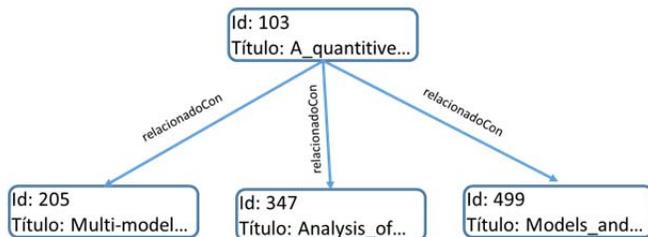


Figura 3. Ejemplo de instancias de la ontología propuesta.

En la ontología propuesta (Fig. 3) se define la clase *Artículo* con dos propiedades de datos: *id* que es su identificador de artículo y *título* que es el título del artículo. Existe una propiedad de objeto *relacionadoCon*, que tiene

como dominio y rango la clase *Artículo* y es simétrica y transitiva. Es precisamente esa propiedad *relacionadoCon* la que servirá para agregar la información al modelo, proveniente del proceso de cálculo de similitud entre documentos bajo cierto umbral de similitud.

La tabla de similitud entre documentos se normalizó tomando en cuenta el valor más alto de similitud y dividiendo cada registro entre este valor. Así se tienen valores de similitud en escala del 0 al 1.

Para determinar el umbral de similitud sobre el cual decidir si un documento es similar a otro se realizó un análisis de la información (Tabla I), obteniendo un promedio en la cantidad de documentos obtenidos como relacionados en primer grado, así como las relaciones inferidas.

TABLA I
SIMILITUDES ENTRE TODOS LOS DOCUMENTOS DE LA COLECCIÓN

umbral	Documentos con relaciones en 1er grado	Promedio de documentos relacionados en 1er grado	Documentos con relaciones en 2do grado	Promedio de documentos relacionados en 2do grado
0.1	797	30.61	797	277.83
0.2	733	5.65	704	17.59
0.3	501	2.34	339	3.61
0.4	287	1.51	111	1.81
0.5	142	1.28	38	1.65
0.6	67	1.16	10	1
0.7	34	1.05	2	1
0.8	22	1	0	0
0.9	16	1	0	0
1	10	1	0	0

El umbral seleccionado es 0.3 que determina que el 62.8% de los documentos tienen en promedio 2 artículos relacionados en primer grado y 4 artículos cuyas relaciones son inferidas. La razón de seleccionarlo es debido a que la cantidad de artículos obtenidos es pequeña y esto ayuda ya que en la experimentación es necesario leer los artículos y determinar la similitud entre ellos por parte de los participantes.

Una vez creada la ontología y seleccionado el umbral se similitud, se agrega información obtenida de la tabla de similitudes al modelo de la ontología. Se requiere de una interfaz para establecer la comunicación entre la aplicación Web y la ontología.

Apache Jena es un framework gratuito, de código abierto, escrito en Java, para la construcción de la Web Semántica [19]. Para nuestra propuesta se utilizará la API de inferencia para obtener los documentos relacionados en segundo grado por medio de la propiedad transitiva *relacionadoCon*.

De esta forma termina el proceso fuera de línea de la propuesta. Ahora conocemos las relaciones de similitud en un primer grado de cada documento especificadas mediante su propiedad *relacionadoCon* y por medio del motor de nuestra ontología, realizando una inferencia de datos, podemos saber con qué otros documentos está relacionado cada documento en un segundo grado, utilizando la propiedad transitiva (Fig. 4).

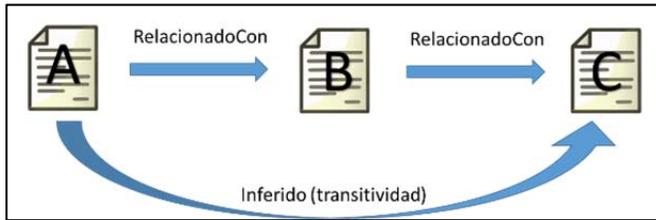


Figura 4. Relaciones de similitud directa e inferida.

La inferencia hace que cada elemento tenga más valor, ya que puede tener efecto en la creación de nueva información [20]. La inferencia se refiere al proceso abstracto de obtener información adicional y un razonador es un objeto de código específico que realiza esta tarea [21].

IV. IMPLEMENTACIÓN

Para la implementación del modelo se utilizó el lenguaje de programación PHP y el manejador de base de datos MySQL. Ambos son de libre distribución y multiplataforma [22]. La Fig. 5 muestra la pantalla principal del sistema que presenta una interfaz sumamente sencilla de un buscador.



Figura 5. Página principal del sistema.

La siguiente captura de pantalla (Fig. 6), muestra el resultado del módulo de búsqueda y recuperación. El usuario proporciona palabras clave y el sistema recupera de la colección de artículos los más relevantes con respecto a la consulta y se los presenta en pantalla.



Figura 6. Página de búsqueda y recuperación.

Por último se presenta la captura de pantalla (Fig. 7) del módulo de recomendación cuando el usuario selecciona algún documento. En este momento se presenta una lista de documentos relacionados en primer grado ordenados de acuerdo a la similitud entre éstos y el documento seleccionado.

También se puede observar (Fig. 7) una lista de documentos inferidos obtenidos de realizar una inferencia en

nuestra ontología y ordenados de acuerdo a la similitud entre éstos y el documento seleccionado. En caso de que algún documento ya ha sido consultado anteriormente por el usuario, éste se presenta en color rojo, con la etiqueta “consultado”.

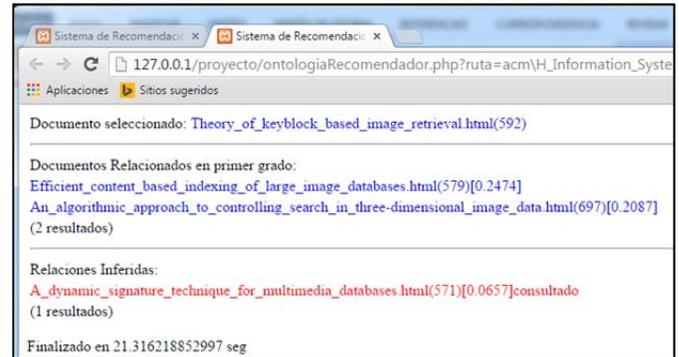


Figura 7. Página del resultado del módulo de recomendación.

V. EXPERIMENTACIÓN

En este apartado se presenta una prueba realizada para valorar el funcionamiento del modelo, basándose en la percepción del usuario. En primera instancia se presenta el objetivo, seguidamente se explica cómo fue desarrollada la prueba y finalmente se dan a conocer las limitaciones y los resultados obtenidos.

A. Objetivo

El objetivo del estudio es recomendar documentos con contenido similar a los que el sistema recupera del repositorio, con base en una consulta del usuario. Se determina el grado de similitud entre el documento recuperado de la consulta y los documentos que el modelo propone como, relacionados en primer grado (aquellos que tienen contenido similar con respecto a los recuperados) y los inferidos (aquellos que tienen contenido similar con respecto a los relacionados en primer grado).

B. Diseño del estudio

Se utilizó como repositorio, 798 documentos digitales obtenidos de la biblioteca digital de la ACM (Association for Computing Machinery), los cuales mantienen una clasificación temática por expertos (directorios). Son archivos con formato *.html*, que contienen los resúmenes de los artículos a los que hacen referencia, con contenido de sólo texto.

Se realizó un estudio donde los sujetos tenían que utilizar el sistema de recomendación realizando consultas relacionadas a un índice de temas proporcionado, correspondiente a las clasificaciones (directorios) del repositorio construido, con el fin de comprobar si el sistema recuperaba documentos relevantes a la consulta y a su vez recomendaría documentos similares al recuperado.

C. Sujetos

Para realizar la prueba se apoyó de un grupo de 7 profesores investigadores expertos en el área de Ciencias de la Computación de la Facultad de Matemáticas de la Universidad Autónoma de Yucatán en México, a los cuales se les indicó una lista de tareas a realizar y se registraron los resultados obtenidos. Posteriormente esos datos se procesaron, se sacó

una media aritmética de los resultados y con ello se obtuvo la evaluación del sistema de recomendación.

D. Procedimiento

Para el experimento, se diseñó un cuestionario de 7 preguntas, 2 de ellas sólo para registro de la información y 5 de ellas para determinar similitudes, las cuales se contestan con una escala Likert de 5 opciones, dónde 1 corresponde a mala y 5 corresponde a excelente.

A continuación se describen las tareas solicitadas al usuario para evaluar el sistema de recomendación:

1. Realizar una consulta al sistema relativa a su área de investigación de acuerdo al índice de temas proporcionado.
2. Seleccionar un documento que sea de su interés.
3. Contestar las siguientes preguntas:
 - a) ¿Cuáles fueron las palabras clave de su consulta?
 - b) ¿cuál es el id del documento seleccionado?

Para las siguientes preguntas, utilizar la escala:

1. Mala	2. Pobre	3. Razonable	4. Buena	5. Excelente
---------	----------	--------------	----------	--------------

- c) ¿Cuál considera que es la similitud entre la consulta y el documento seleccionado?
 - d) ¿Cuál considera que es la similitud entre la consulta y el primer relacionado?
 - e) ¿Cuál considera que es la similitud entre la consulta y el primer inferido?
 - f) ¿Cuál considera que es la similitud entre el documento seleccionado y el primer relacionado?
 - g) ¿Cuál considera que es la similitud entre el documento seleccionado y el primer inferido?
4. Se pidió repetir el proceso en más de una ocasión sin limitarlos a un número máximo de consultas.

E. Resultados

Se analizaron las 19 consultas hechas al sistema en total por todos los participantes y se obtuvo la siguiente información:

Como se puede observar en la Fig. 8, en el caso de la similitud entre la consulta hecha al sistema y el documento seleccionado se obtuvo un promedio de 2.8 con una desviación estándar de 1.33. Para la similitud entre la consulta y el primer documento relacionado se obtuvo una media de 2.58 con una desviación estándar de 1.17 mientras que para la similitud entre la consulta y el primer documento inferido se registró una media de 2.26 con una desviación estándar de 1.24.

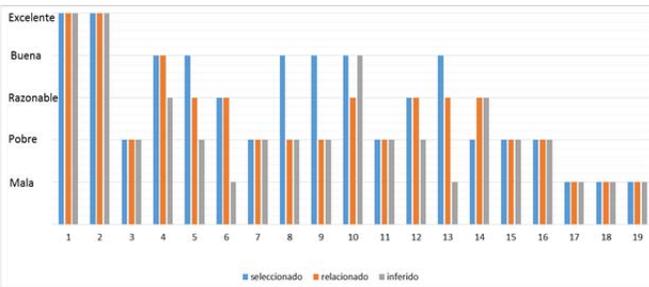


Figura 8. Similitud entre las consultas y los documentos seleccionados y recomendados.

En el caso de la similitud entre el documento seleccionado y el primer documento relacionado se obtuvo un promedio de 3.42 con una desviación estándar de 0.84, mientras que para la similitud entre el documento seleccionado y el primer documento inferido se registró una media de 2.89 con una desviación estándar de 1.24 (Fig. 9).

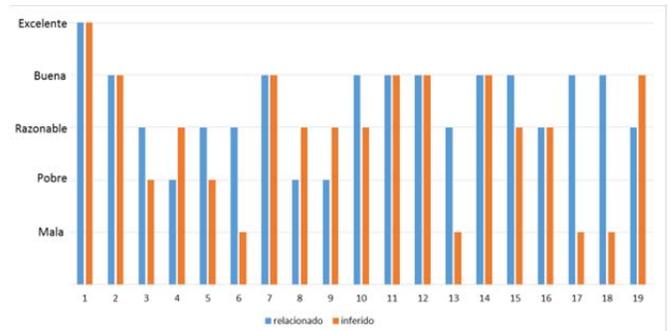


Figura 9. Similitud entre el documento seleccionado y los documentos recomendados.

VI. CONCLUSIONES

Con los resultados de la experimentación podemos concluir que la metodología presentada para el cálculo de similitud entre documentos funciona aceptablemente con 84.21% de respuestas de similitud de razonable a excelente entre el documento seleccionado y el primer documento relacionado proporcionado por el sistema; mientras que la similitud entre el documento seleccionado y el primer documento inferido obtuvo un 68.42% de respuestas de razonable a excelente, lo cual era lo esperado, ya que los documentos inferidos tienen una similitud menor al umbral seleccionado.

Por otro lado, se pudo analizar que para la similitud entre los documentos y la consulta, el promedio para los tres casos (seleccionado, relacionado e inferido) estuvo por debajo de 3 (razonable), lo que nos indica que la búsqueda y recuperación, no está funcionando como se esperaba dentro de la arquitectura presentada. Esto se debe a las siguientes situaciones:

1) El usuario espera obtener como respuesta del sistema de búsqueda y recuperación las palabras claves que él proporcionó en el título del artículo.

El sistema indexa los términos del contenido de los resúmenes de los artículos pero no toma en cuenta los términos del título ni los términos de su clasificación.

2) El usuario espera que proporcionando en su consulta términos muy específicos como por ejemplo “np” o “3D” obtenga resultados específicos.

El sistema, si bien indexa este tipo de términos menores a 3 letras, la función *match* de mysql utilizada para recuperar la información, no los toma en cuenta por defecto, sin embargo es posible cambiar esa configuración.

Como trabajo futuro inmediato se propone incluir los términos del título del artículo así como los de su clasificación en el vector de términos de cada documento. Se podría además incluir las palabras clave del artículo, sin embargo ese dato no se tiene el momento.

Se pretende incluir la parte semántica al proceso, utilizando un tesoro de sinónimos para los términos que representan a cada documento.

También se tiene contemplado replicar el experimento incrementando el número de participantes para corroborar los resultados obtenidos.

Para la presentación del sistema, se propone mejorar la interfaz (diseño), así como mostrar el resumen del artículo en la parte del sistema de recuperación, cuando el usuario coloque el puntero sobre el título del mismo.

Así como analizar y optimizar el proceso de comunicación entre la aplicación web y la API de la ontología (java), para reducir el tiempo de espera de resultados entre estos dos procesos.

REFERENCIAS

- [1] Internet world stats. <http://www.internetworldstats.com/stats.htm>. Consultado el 12 de mayo de 2014.
- [2] Olivas, J. Búsqueda eficaz de información en la Web, Editorial de la Universidad Nacional de la Plata (Edulp). Buenos Aires, Argentina. (2011).
- [3] Álvarez, C., Detección de Similitud Semántica en Textos Cortos. Tesis de Maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica. Tonantzintla, Puebla, (2014).
- [4] Manning D., Schütze, H., Foundations of stational natural language processing. MIT press, (1999).
- [5] Verspoor K., Cohen K., Natural Language Processing. Encyclopedia of Systems Biology, p. 1945 – 1948, (2013).
- [6] Baeza-Yates, R. A. y Ribeiro-Neto, B., *Modern Information Retrieval*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., (1999).
- [7] Manning, C., R Raghavan, P. y Schütze H., *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England, (2009).
- [8] Olivas, J., *Búsqueda eficaz de información en la Web*, Editorial de la Universidad Nacional de la Plata (Edulp), Buenos Aires, Argentina, (2011).
- [9] W3C. <http://www.w3.org/standards/semanticweb/>. Consultado el 10 de Julio de 2015.
- [10] Gruber, T. R., "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" en *International Journal of Human and Computer Studies*, 43 (5-6), p. 907-928. (1995).
- [11] Mclean, N., Lynch, C. Interoperability between Information and Learning Environments: Bridging the Gaps. A Joint White Paper on behalf of the IMS Global Learning Consortium and the Coalition for Networked Information. DRAFT – Versión de Junio 28, (2003). www.imsglobal.org/DLims_white_paper_publicdraft_1.pdf
- [12] Salton, G., *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison Wesley, (1989).
- [13] <https://dev.mysql.com/doc/refman/5.5/en/fulltext-natural-language.html>. Consultado el 1 de junio de 2015.
- [14] Salton, G. y Yang, C.S. On the specification of term in automatic indexing. *Journal of Documentation*, 29 (4), pp. 351-72, (1973).
- [15] Korfhage, R. "Information storage and retrieval", Wiley, New York. (1997).
- [16] Salton, G., Buckley, C. Term-weighting approaches in automatic text retrieval. *Information & processing managemet*, 24(5), pp. 513-23. (1988).
- [17] Protégé. The Protégé Project. <http://protege.stanford.edu>. Consultado el 25 de Mayo 2015. (2000).
- [18] Noy, N. F., y McGuinness, D. L. Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología, 1–29. (2005).
- [19] Apache Jena. <https://jena.apache.org/>. Consultado el 25 de Mayo 2015.
- [20] Yu, L. Introduction to the Semantic Web and Semantic Web Services. Chapman y Hall. (2007).
- [21] Reasoners an rule engines: Jena inference support. <https://jena.apache.org/documentation/inference/index.html>. Consultado el 24 de mayo 2015.
- [22] Lemire D. y McGrath S., "Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL", Technical Report D-01, (2005).



Rodrigo Vences-Nava received his degree in Computer Science from Autonomous University of Yucatán in 2005. He works for the same institution since 2007 as Information Technology Management. His current research interest are recommender systems, digital theses, and data mining.



assisted living.

Jorge Gomez-Montalvo is Lecturer and Researcher at the Mathematics Faculty of the Autonomous University of Yucatan, Mexico. He holds a Ph.D. in Computer Science from the National Institute of Applied Sciences of Toulouse, France. His research interests include autonomous and semantic frameworks for distributed services, and its applications for developing new technologies in ambient



Víctor Menéndez-Domínguez is Lecturer and Researcher at the Mathematics Faculty of the Autonomous University of Yucatan, Mexico. He holds a Ph.D. in Advanced Informatics Technologies from the University of Castilla-La Mancha, Spain. His research interests include knowledge management, repositories and metadata, information retrieval.