Inteligencia Artificial 19(57), 1-21
doi: 10.4114/ia.v18i56.1038

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Organización de documentos mediante grafos de relaciones semánticas

Santa Vallejo Figueroa[1] y Valeria Nava Lozano[2]
[1] Instituto Tecnológico Superior de Poza Rica (México) / Universidad Nacional de Educación a Distancia. Departamento de Lenguajes y Sistemas (España) svallejo36@alumno.uned.es
[2] Universidad de Guadalajara, México. vnava@teachers.org

Resumen. Actualmente los documentos son el principal medio para representar información en varios dominios. Continuamente los usuarios almacenan documentos en discos duros o unidades de almacenamiento en línea siguiendo una organización personal basada en temas, pero los documentos pueden contener varios temas. Esto dificulta el acceso a los documentos cuando el tema deseado no corresponde con el almacenado. Básicamente los actuales motores de búsqueda de documentos se basan en el nombre del archivo o palabras clave del contenido, donde el término o términos a buscar debe coincidir exactamente con el nombre o contenido. En este artículo se propone un método para organizar documentos mediante grafos, teniendo en cuenta los temas que los documentos contienen. Para ello se genera un grafo por cada documento considerando los sinónimos, términos relacionados semánticamente, hipónimos e hiperónimos de los sustantivos y verbos que se encuentran en los documentos. La propuesta ha sido comparada contra las herramientas *Google Desktop* y *LogicalDoc* con buenos resultados.

Palabras clave: Recuperación de información, organización de documentos, relaciones semánticas, grafos.

Abstract. Nowadays documents are the main mean for representing information in several domains. Continuously users store documents on hard disks or online storage services according to some personal organization based on topics, but such documents can contain one or more topics. This situation makes hard to access documents when the desired topic does not match with the one stored. Basically current search engines search based on the filename or keywords from content, but the desired term or terms must match exactly as are in the filename or content. In this paper, a method for organizing documents by means of graphs is proposed, taking into account the topics of documents. For this a graph for each document is generated taking into account synonyms, semantic related terms, hyponyms, and hypernyms of nouns and verbs contained in documents. The proposal has been compared against *Google Desktop* and *LogicalDoc* tool with good results.

Keywords: Information retrieval, document organization, semantic relations, graphs.

1 Introducción

Hoy en día gran cantidad de información se guarda en documentos (ficheros de texto) pues constituyen el medio más común para capturar datos, información y, consecuentemente, conocimiento. Su uso incluye a todos los dominios de la vida moderna, desde documentos personales hasta organizacionales. Incluso, en algunos dominios los documentos constituyen la piedra angular para el desempeño de actividades, tal es el caso del sector bancario y médico.

ISSN: 1988-3064(on-line)

El aumento de documentos en discos duros de equipos personales o unidades de almacenamiento en línea exige una adecuada organización para encontrar y recuperar documentos de manera ágil. Usualmente para almacenar documentos los usuarios adoptan sus propios esquemas de organización (laboral, personal, ocio, temas, fechas, etc.) tomando en cuenta aspectos que les ayudan a agrupar documentos. Sin embargo esto se reduce a una ubicación física o a un nombre significativo del fichero; esta forma de organizar los documentos requiere que los usuarios conozcan la ubicación de cada documento, tengan idea de cuándo fueron creados o al menos el nombre o extensión del fichero. Sin embargo, los nombres de fichero o rutas no proporcionan ayuda significativa al realizar búsquedas y las herramientas existentes no facilitan la gestión y recuperación de la información [1]. Es deseable contar con algún mecanismo que permita organizar documentos tomando en cuenta la información que contienen, dejando de lado la ubicación, el nombre del fichero, la fecha de creación y cualquier esquema de organización subjetiva. De tal manera que el usuario pueda recuperar documentos únicamente consultando la idea que tiene el usuario en mente de lo que desea encontrar. Este reto de organización de documentos se contextualiza en el domino de búsquedas de escritorio, donde el usuario almacena documentos en equipos de cómputo personales o sistemas centralizados de almacenamiento en línea.

El ciclo de vida básico de los documentos se compone de creación, guardado, consulta, actualización y eliminación. Si bien las etapas de creación y guardado son controlables por el usuario, la consulta se puede complicar cuando se manipulan muchos documentos. En consecuencia, las etapas de actualización y eliminación pueden no llegar a realizarse. Para ayudar al usuario en la búsqueda de documentos existen algunas herramientas básicas propias de sistemas operativos (*find*, *grep*, etc.); también las hay con más utilidades como *Launchy*[1], *Copernic*[2], *X1*[3], *Alfresco*[4], *Quicksilver*[5], etc., o algunos sistemas más sofisticados como los sistemas de gestión documental y sistemas de gestión de contenidos, que incluyen etapas adicionales como distribución, control de acceso, seguridad, publicación, etc. En general los sistemas básicos y más sofisticados se caracterizan porque su búsqueda se basa en el nombre del fichero o en palabras clave de su contenido. Estos sistemas normalmente ejecutan búsqueda léxica, no semántica. Esto significa que para buscar un documento se debe elegir la palabra o parte de una palabra que coincida exactamente en el nombre del fichero, del directorio donde se encuentra o con alguna palabra clave de su contenido. Sin embargo, un usuario típico hace búsquedas escribiendo palabras de la idea que tiene en mente, sin pensar cómo se llama el fichero, en qué ubicación está o cuáles son las palabras clave vinculadas a lo que desea. Esta problemática se acrecienta si existe gran cantidad de documentos en el repositorio en donde busca el usuario. Además hay que tener en cuenta que los documentos se almacenan utilizando diferentes formatos y podría tener diferentes formas de ser organizados. Esto hace que algunos documentos importantes sean descartados en la búsqueda.

Una solución a la situación anteriormente planteada es organizar los documentos con base en la información semántica de su contenido. Este enfoque, al estar basado en el contenido de los documentos, elimina la subjetividad del usuario para organizar los documentos. La semántica de un documento se refiere a las ideas que se desean transmitir mediante su contenido. Una manera de obtener la semántica es identificar los sustantivos más relevantes y sus relaciones semánticas, las cuales pueden ser sinónimos, merónimos, hipónimos, hiperónimos, etc. Estas relaciones semánticas permiten que los sustantivos relevantes puedan agruparse para identificar los temas de un documento.

En este artículo se describe un método para organizar documentos escritos en Inglés, el cual está basado en grafos para relacionar documentos de acuerdo a los temas que contienen. El método se basa en los sinónimos, términos semánticamente relacionados, hipónimos e hiperónimos de los sustantivos de las *relaciones sustantivo-verbo* más relevantes de los documentos. De tal manera que el usuario consigue una organización dinámica y personalizada de sus documentos. El resto del artículo está organizado de la siguiente manera. La Sección 2 presenta una breve reseña de los trabajos relacionados. La Sección 3 describe la estrategia para organizar y buscar documentos mediante grafos. La Sección 4 muestra la evaluación y comparación de la implementación del método propuesto. Por último, la Sección 5 contiene las conclusiones.

[1] www.launchy.net [visitado en agosto de 2015]
[2] www.copernic.com [visitado en agosto de 2015]
[3] www.x1.com [visitado en agosto de 2015]
[4] www.alfresco.com [visitado en agosto de 2015]
[5] www.broadvision.com/en/products/quicksilver [visitado en agosto de 2015]

2 Trabajo Relacionado

Desde la perspectiva de organización y búsqueda de documentos con enfoque general se puede ver el trabajo de Price *et al.* [2], quienes describieron un enfoque basado en *componentes semánticos* para la búsqueda de documentos. Estos documentos se caracterizan porque son documentos web de un dominio específico y ya están clasificados. La propuesta se centró en cómo definir los componentes semánticos a partir de los contenidos de los documentos. Toman en cuenta el dominio y tema principal del documento. Definieron un conjunto de clases de documentos y por cada clase tenían asociado un conjunto de componentes semánticos. Este modelado es complementario del contenido y de la indexación. Zhong *et al.* [3] propusieron un esquema de búsqueda semántica haciendo corresponder grafos conceptuales obtenidos a partir del contenido de documentos; los documentos que procesan son obtenidos de la Web (páginas web). Dicho trabajo requiere de antemano una jerarquía de conceptos y una jerarquía de relaciones (entre conceptos); sin los cuales la propuesta no podría funcionar. Una idea muy parecida a nuestro trabajo es la que propusieron Giunchiglia *et al.* [4], pero desde un punto de vista más formal empleando Lógica Descriptiva. Los autores se enfocan a la búsqueda de conceptos en documentos, para lo cual pre-procesan las consultas y obtienen los sentidos de la consulta para hacer la búsqueda de conceptos en una base de conocimiento terminológica. Para ello trasladan las palabras de la consulta a conceptos. Por otro lado, las redes neuronales SOM (*Self-Organized Maps*) [5] han sido aplicadas exitosamente en varios trabajos sobre organización y búsqueda de documentos [6, 7, 8, 9, 10, 11]. Por ejemplo, Fernandes y Ludermir [12] propusieron un método de extracción de características para representación de documentos relacionados mediante redes neuronales auto-organizadas. Esta representación es más compacta que las usadas normalmente (binaria o frecuencia de términos). Esta propuesta es funcional sólo si el modelo de organización es basado en redes neuronales SOM. Salah [13] presentó un método para la personalización de información de búsqueda web. El núcleo del método es una red neuronal SOM semántica que agrupa documentos similares de acuerdo a su contenido, aunque el contenido sólo está dado por el *snippet* que lo representa.

Desde la perspectiva de organización y búsqueda de documentos con énfasis en documentos XML [14, 15] se puede ver el trabajo de Cohen *et al.* [16], quienes presentaron un motor de búsqueda semántica sobre documentos XML. Los autores aprovechan la estructura XML de los documentos para realizar las consultas. La aportación principal de este trabajo es la manera de cómo definir las consultas. Vagena y Moro [17] desarrollaron un método de búsqueda semántica de documentos XML en flujos dinámicos. El método emplea un lenguaje de recuperación propio que no depende de términos sintácticos para buscar documentos en la estructura XML.

Desde la perspectiva de organización y búsqueda de documentos del dominio biomédico se puede ver el trabajo de Nobata *et al.* [18], quienes implementaron un método para búsqueda de documentos de Biomedicina. Dicho método emplea metadatos (conceptos semánticos) para indexar los documentos. Tales metadatos se obtienen a partir del minado del contenido de los documentos empleando una fuente de datos del dominio. Así, el método permite hacer búsqueda de información textual y metadatos sobre MEDLINE. En este mismo sentido, Lourenco *et al.* [19] presentaron un enfoque de indexación semántica para la recuperación de documentos biomédicos en PubMed. La base del método es la identificación de Entidades Nombradas del dominio a partir de los términos relevantes del documento. Para la indexación emplean un espacio bi-dimensional donde los términos relevantes se organizan de acuerdo a la probabilidad de que representen documentos relevantes e irrelevantes. También Oh *et al.* [20] presentaron un método para búsquedas de documentos biomédicos en PubMed. Para ello propusieron un algoritmo que crea redes semánticas sin redundancia. Los autores identifican los términos más representativos y a partir de ahí determinan lo que denominan *superconceptos* (que comprende muchos términos). A partir de *superconceptos* poco frecuentes (realmente relacionados) construyen lo que llaman *red semántica mínima* y luego una *red semántica compacta*, que es la que usan para hacer búsquedas.

Desde la perspectiva de organización y búsqueda de documentos basada en ontologías y tesauros se puede ver el trabajo de Eriksson [21], quien presentó un enfoque que combina documentos y ontologías. Para ello emplea anotaciones en documentos PDF, tales anotaciones las relaciona con los conceptos de una ontología dependiendo el dominio del documento. De esta manera se genera un meta-nivel que asemeja la estructura del documento a la ontología, de tal manera que la ontología puede describir completamente el documento: sus partes, conceptos y frases. Lupiani-Ruiz *et al.* [22] desarrollaron un motor de búsqueda de documentos (noticias) en el dominio de finanzas,

extrapolable a otros dominios. El motor emplea como núcleo a ontologías para organización y búsqueda. A partir de información semi-estructurada y no estructurada obtenida de la Web se crean instancias de la ontología que sirven para mantener actualizado al motor. La búsqueda emplea la estructura de la ontología y anotaciones hechas a la información obtenida de la Web, para lo cual emplean las instancias de la ontología y sus propiedades. Zhang *et al.* [23] definieron un esquema de recuperación semántica de información basado en ontologías. Dicho esquema emplea a) una representación semántica de documentos basada en la ontología, b) un módulo de extensión de consultas para extraer las características semánticas de la consulta y c) un método de recuperación semántica de documentos que emplea consultas jerárquicas, donde se comparan el grafo de la consulta con los grafos de los documentos. Bhagdev *et al.* [24] desarrollaron un método de búsqueda de información que combina búsqueda basada en palabras clave y semántica. Los documentos se indexan mediante palabras clave y las anotaciones sobre los documentos en tripletas RDF. La búsqueda sintáctica se ejecuta sobre la representación de palabras clave, mientras que la búsqueda de metadatos se ejecuta sobre las tripletas RDF. Como resultado se obtienen documentos que resultan de mezclar y hacer corresponder los resultados de ambas búsquedas. Uno de los trabajos pioneros en el uso de WordNet[6] para organización de documentos fue realizado por Gonzalo *et al.* [25], donde se emplearon synsets y sentidos de WordNet para enriquecer la representación de términos en la indexación de documentos; con lo que se logra desambiguar los términos y que se puedan identificar términos relacionados. Reforgiato [26] empleó WordNet para la organización de documentos (mediante agrupación) después de recuperarlos. Para la representación de texto se emplea información léxica y la ontología de WordNet, lo que permite que se obtenga una representación de baja dimensionalidad. En este mismo sentido, Chen *et al.* [27] emplearon WordNet para la representación de documentos. Con base en el contenido de los documentos, los autores identifican los términos relevantes, los cuales organizan jerárquicamente empleando hiperónimos de WordNet. Esta organización jerárquica les permite identificar temas relacionados. Dragoni *et al.* [28] también emplearon WordNet para la representación de documentos y consultas sobre los mismos. Para reducir la redundancia de la información emplean la ontología de WordNet aprovechando las relaciones jerárquicas *is-a* de los sustantivos, no se toman en cuenta verbos, adverbios ni adjetivos. Lo anterior les permite trabajar sobre conceptos en lugar de términos.

Las diferencias de los trabajos anteriores respecto a la propuesta que se presenta en este artículo son las siguientes: a) para la representación del texto se emplea una variante del Modelo Espacio Vectorial (MEV) pero en lugar de representar términos aislados se representan relaciones *sustantivo-verbo*, con lo que se obtiene una representación de baja dimensionalidad; b) para la identificación de temas se obtienen los conceptos más relevantes; c) los temas de un documento se representan mediante grupos de sustantivos semánticamente relacionados, tales grupos denotan conceptos; d) mediante relaciones semánticas de sinonimia, hiponimia e hiperonimia se relacionan temas dentro un mismo documento y entre documentos; e) los documentos que se procesan no tienen estructura alguna, son de texto plano no etiquetado, lo que permite que los documentos sean de cualquier dominio y f) no se emplean diccionarios, tesauros, vocabularios ni ontologías para la identificación de temas.

3 Propuesta

En general, la propuesta de organización de documentos consiste en: 1) una representación de baja dimensionalidad que se basa en *relaciones sustantivo-verbo*, 2) la agrupación de sustantivos en conceptos para denotar los temas de un documento y 3) la identificación de relaciones entre temas (dentro del mismo documento y entre diferentes documentos) mediante relaciones semánticas de sinonimia, hiponimia e hiperonimia. Con lo anterior se construyen grafos por cada documento, los cuales se estructuran tomando en cuenta las relaciones semánticas. A su vez, los grafos se relacionan con otros con los que compartan sustantivos formando un meta-grafo (grafo de grafos).

La base del método propuesto es el modelo de representación de texto que se emplea. Este modelo es una extensión del modelo de *matrices palabra-documento* propuesto por Deerwester *et al.* [29] y Turney [30], el cual es derivado del *Modelo Espacio Vectorial* propuesto por Salton, et al. [31]. A diferencia de la idea principal de las *matrices palabra-documento*, que consideran la frecuencia de palabras en los documentos, el modelo propuesto en este trabajo extiende

[6] wordnet.princeton.edu [visitado en agosto de 2015]

esa idea pero toma en cuenta la frecuencia de aparición de *relaciones sustantivo-verbo* dentro de un documento. A diferencia de otro tipo de relaciones (*es-un, pertenece-a, es-parte-de*, etc.), se seleccionan *sustantivos* relacionados a *verbos* debido a que en el contexto en el que se encuentran dichas relaciones contribuyen a determinar la idea que se desea transmitir en una frase. Las *relaciones sustantivo-verbo* representan las principales interacciones en las frases; donde el papel principal recae en los sustantivos (*sujeto*) que ejecutan una acción (*verbo*). Por sí sola una *relación sustantivo-verbo* relevante en una frase denota el significado semántico de dicha frase, de manera similar a como un usuario forma sus ideas en la mente. Esto se justifica mediante la *Hipótesis Distribucional* de Harris [32], que indica que palabras en un mismo contexto denotan un mismo significado; ese significado es resumido por las *relaciones sustantivo-verbo*. Cuando se agrupan los actores principales -*sustantivos*- de las frases de un documento (extraídos de las *relaciones sustantivo-verbo*) se pueden identificar los conceptos y en consecuencia los temas que trata dicho documento. Dado que se identifican las *relaciones sustantivo-verbo* de un documento y los temas del documento, el modelo de representación propuesto captura la semántica de las frases de un documento, por lo que facilita la búsqueda de la idea que tiene un usuario cuando realiza una consulta. Así pues, el modelo de representación propuesto refuerza la utilidad de las *matrices palabra-documento*, pero no para determinar la similitud entre palabras, sino para capturar las ideas de documentos y mediante éstas organizar tales documentos.

El trabajo realizado puede resumirse mediante la descripción de los dos módulos generales que se han implementado (ver Figura 1): organización y búsqueda. El módulo de organización se ocupa de representar, indexar y estructurar la información de los documentos, mientras que el módulo de búsqueda se encarga de procesar las consultas calculando la similitud de los documentos respecto a la consulta. Ambos módulos se explican a continuación.

3.1 Organización

Este módulo general se compone de 6 módulos específicos que se explican a continuación.

3.1.1 Extracción de texto

En este módulo se extrae el texto que contienen los documentos, el cual se guarda como texto plano. Los documentos pueden provenir de los formatos TXT, PDF, DOC y DOCX. Esta extracción no incluye información contenida en metadatos, imágenes o tablas que se encuentren en el documento. Para esto se emplea la herramienta Apache Tika toolkit[7].

[7] tika.apache.org [visitado en agosto 2015]

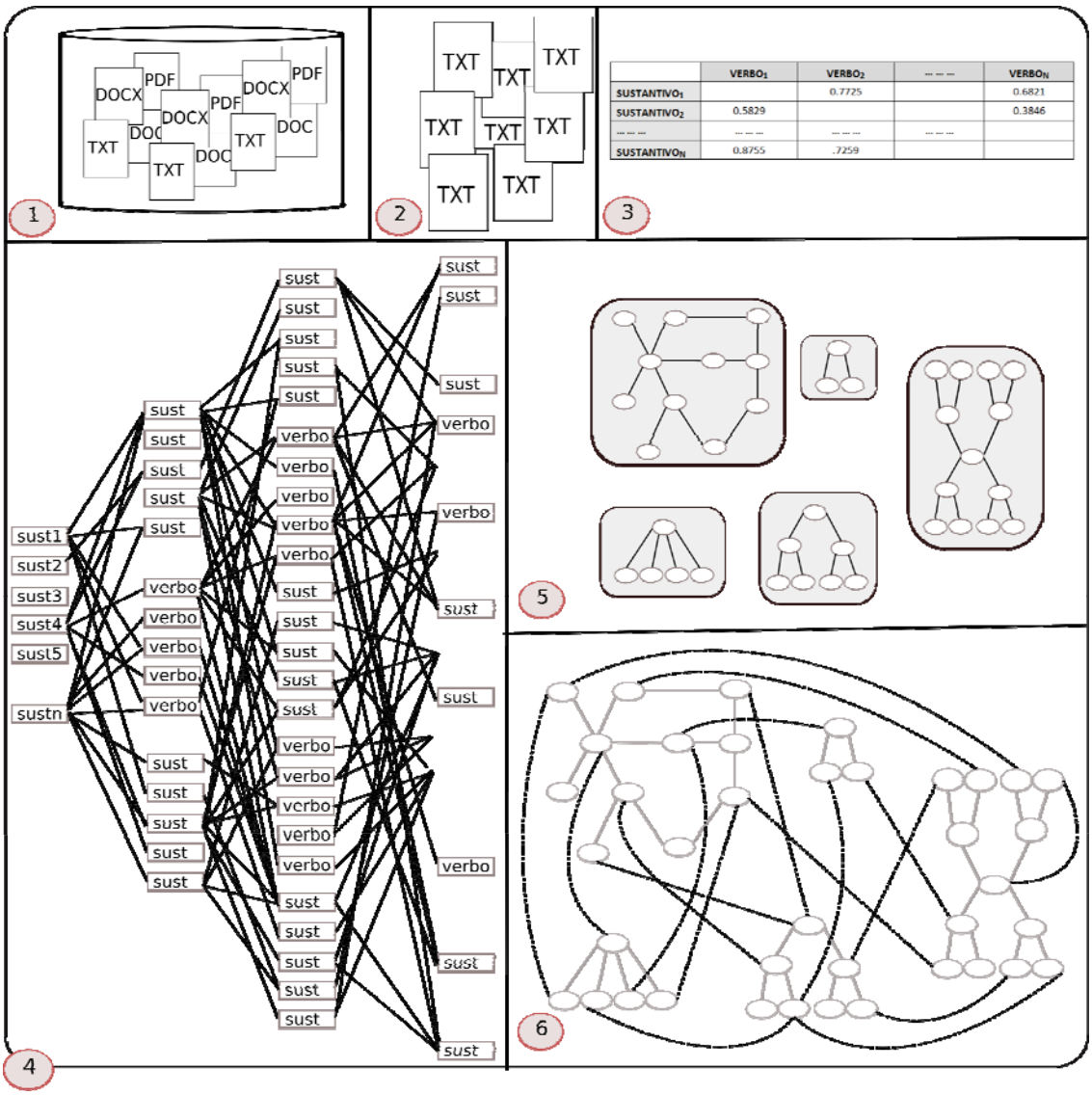

Figura 1. Propuesta de organización de documentos

3.1.2 Preprocesamiento de texto

Aquí se obtiene la categoría gramatical (sustantivo, verbo, adjetivo, adverbio, etc.) mediante la herramienta TreeTagger[8]. Posteriormente se hace un filtrado del texto original para eliminar *palabras vacías* (independientemente de su categoría gramatical), es decir, aquellas que no aportan información significativa al contenido del documento, como por ejemplo artículos, preposiciones, conjunciones, pronombres, etc. Para realizar esto se emplea código propio y una lista de palabras vacías.

[8] www.cis.uni-muenchen.de/~schmid/tools/TreeTagger [visitado en agosto 2015]

3.1.3 Representación de texto

Una vez que se ha filtrado el texto lo siguiente es representarlo de manera manejable por la computadora. Para ello sólo se toman en cuenta los sustantivos y verbos que aparecen en el texto. No se toman en cuenta a todos los sustantivos, lo que interesa es identificar aquellos sustantivos que juegan un papel determinante en las frases. De tal manera que se identifican las *relaciones sustantivo-verbo*; es decir, cuando un sustantivo ocurre junto a un verbo. Esto se debe a que los sustantivos vinculados a un verbo dan idea de que un ente realiza algo, lo cual refleja la idea de la frase. Para ello se genera una matriz de contingencia que denota la ocurrencia de un sustantivo con los diferentes verbos con los que está relacionado en el texto. De las *relaciones sustantivo-verbo* se obtienen los lexemas para ambos, quedando una relación de *lexema-de-sustantivo – lexema-de-verbo*[9]. Las *relaciones sustantivo-verbo* se representan siguiendo la idea de las *matrices palabra-documento* del *Modelo Espacio Vectorial* (MEV) tomando en cuenta el peso de dichas relaciones, el peso se calcula mediante *TF-IDF* [33]. Lo que se obtiene de este módulo es una ponderación de las *relaciones sustantivo-verbo* más repetidas; es decir, aquellos *sustantivos* que están involucrados en frases que expresan acciones (*verbos*). Sólo se consideran las *relaciones sustantivo-verbo* con más alta ponderación TF-IDF, superior al 50%. Este umbral se estableció (de acuerdo al Teorema del Límite Central [34]) mediante inspección manual de 500 experimentos de representación de documentos (del dataset Reuters-21578[10]) para conocer qué tan relevantes eran las relaciones en los documentos. Aquellas relaciones que tienen una ponderación menor al umbral establecido aportan poco, muy poco o nada a los temas del documento y, negativamente, hacen que el algoritmo de búsqueda sea computacionalmente más caro. Para realizar esto se emplea código propio.

3.1.4 Identificación de temas

No sólo basta identificar qué pares *sustantivo-verbo* son los mejor ponderados dentro del texto, sino que es deseable agrupar aquellos pares que tienen alguna relación con otros. Esta agrupación puede hacerse extrayendo el sustantivo de las *relaciones sustantivo-verbo* identificadas en el paso previo. Agrupar los sustantivos extraídos permite identificar los *conceptos* de un documento. A partir de los conceptos se pueden identificar los temas que trata un documento [35]. Para esto se emplea el algoritmo de agrupación CBC (*clustering by committee*) [36]. Mediante este algoritmo se agrupan los sustantivos, cada grupo denota un concepto[11]. A partir de los sentidos de los sustantivos en cada grupo se determina el sentido ganador, el cual es asociado al concepto que representa al grupo. Así, el tema principal de un texto corresponde al concepto que representa al grupo más grande, el segundo tema al segundo grupo más grande y así sucesivamente. Debido a que los sustantivos de los conceptos identificados corresponden a *relaciones sustantivo-verbo*, por cada sustantivo se puede acceder al verbo con el que está relacionado, de tal manera que por cada tema del texto se puede conocer las acciones que ejecutan los sustantivos del tema. Para realizar esto se emplea código propio.

3.1.5 Identificación de relaciones entre temas

De acuerdo con lo anterior, un documento puede modelarse por los temas (*conceptos* más importantes) que describen las *relaciones sustantivo-verbo* que contiene. Si bien dichas clases permiten organizar por temas a los documentos, es deseable identificar la relación que tienen diversos documentos si tratan sobre temas similares. Las relaciones entre temas se identifican a partir de las relaciones que tienen los sustantivos-verbos que componen un tema con los sustantivos-verbos de otros temas. Para ello se identifican las relaciones semánticas que tiene cada sustantivo que componen un tema. Estas relaciones semánticas son: a) *sinónimos*, b) *términos semánticamente relacionados* -sustantivos y verbos-, c) *hipónimos* y d) *hiperónimos*. Para esto se emplean las herramientas DISCO[12] y WordNet.

---

[9] En el resto del artículo esta relación se denomina únicamente relación *sustantivo-verbo* pero debe entenderse que realmente son los lexemas a los que se está haciendo referencia.

[10] www.daviddlewis.com/resources/testcollections/reuters21578 [visitado en agosto 2015]

[11] Pantel [36] y Lin y Pantel [37] denominan *clase semántica* a un *concepto*, el cual es el representante de un grupo de sustantivos. Los sustantivos de cada grupo tienen la característica de ser semánticamente similares entre sí. La similitud semántica de los sustantivos se calcula mediante el algoritmo CBC. Por ejemplo, el concepto *medio-de-transporte* agrupa a los sustantivos *automóvil, autobús, barco, yate, avión* y *motocicleta*, los cuales en común tienen el sentido de que se usan como medios de transporte. El sentido preponderante de los sustantivos de un concepto denotan el tema del que trata dicho concepto.

[12] www.linguatools.de/disco/disco_en.html [visitado en agosto 2015]

3.1.6 Generación del grafo de relaciones semánticas

A partir de las relaciones semánticas identificadas en el punto anterior se construye un grafo de relaciones semánticas para modelar cada tema de cada documento. De tal manera que los grafos de todos los documentos se relacionan unos con otros de acuerdo a las *relaciones sustantivo-verbo* (identificadas en el punto anterior) que se tienen entre documentos. El grafo de un documento se modela de la siguiente manera. Cada sustantivo de las *relaciones sustantivo-verbo* de cada tema corresponderá a un *nodo*, dentro de dicho nodo hay: a) un identificador del nodo, b) el sustantivo, c) el verbo con el que se relaciona el sustantivo y d) una lista de los documentos en los que ocurre el sustantivo con el verbo. Cada nodo estará conectado con otros nodos (también sustantivos) que representan los a) sinónimos, b) términos semánticamente relacionados, c) hipónimos y d) hiperónimos del sustantivo, tal como ilustra la Figura 2. Para realizar esto se emplea código propio.

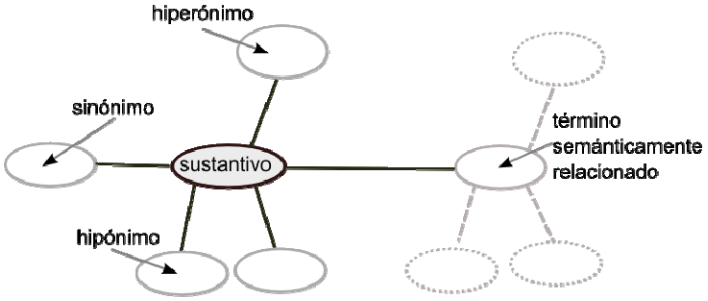

Figura 2. Relaciones semánticas que puede tener un sustantivo

A los *arcos* se les asigna un peso de acuerdo a la relación que conecta (sinónimo, término semánticamente relacionado, hipónimo o hiperónimo) con el sustantivo. Si la relación entre un sustantivo y otro ocurre muchas veces en todos los documentos entonces el peso de la relación (arco) se multiplica por el número de ocurrencias que se den. De esta manera, a cada tipo de relación se le asigna una ponderación numérica que denota su relevancia hacia el sustantivo. Los valores empleados para los pesos son 1.0, 0.9, 0.8 y 0.7 respectivamente para los sinónimos, términos semánticamente relacionados, hipónimos e hiperónimos. Estos valores se definieron considerando la posición que tienen en el grafo respecto al sustantivo al que conectan. Un sinónimo es la relación más importante (valor máximo) ya que significa lo mismo que el sustantivo al que se conecta, un término semánticamente relacionado es menos importante que un sinónimo debido a que dicho término da idea del sustantivo al que se conecta, pero no significa lo mismo; un hipónimo es menos importante que un término semánticamente relacionado ya que denota que depende del sustantivo al que se conecta, de tal manera que el sustantivo generaliza la idea del hipónimo; el menos importante es un hiperónimo debido a que es una generalización del sustantivo al que se conecta.

Después de llevarse a cabo los 6 módulos anteriormente descritos, el grafo construido puede verse desde dos perspectivas (ver Figura 3):

1. Donde se encuentran relacionados los sustantivos y verbos más importantes de un conjunto de documentos, y por éstos, los documentos en donde ocurren. Es decir, de lo más granular a lo más general.

2. Donde se encuentran entrelazados los documentos de acuerdo a las relaciones de sinonimia, término semánticamente relacionado, hiponimia o hiperonimia de los sustantivos que contienen. Es decir, de lo más general a lo más granular.

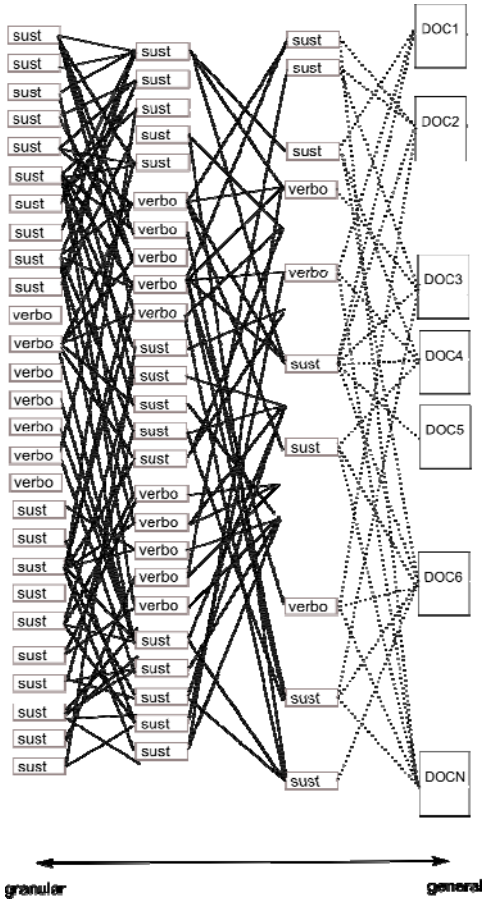

Figura 3. Relaciones entre documentos y sustantivos-verbos

3.2 Búsqueda

Una vez obtenido el grafo de relaciones semánticas se puede buscar por un sustantivo en particular y recuperar los documentos que estén relacionados con dicho sustantivo. Una vez identificado el sustantivo deseado se obtienen los nodos con los que está relacionado y a partir de dichos nodos se obtienen los documentos asociados. Es decir, accediendo a un nodo del grafo se puede llegar a otros nodos siempre y cuando exista alguna relación semántica entre los nodos.

La búsqueda en el grafo se realiza mediante consultas. Las consultas que se formulen deben contener el *sustantivo*, *verbo*, *relación sustantivo-verbo* o *frase* deseados. Como resultado se obtiene un listado de documentos ponderados de acuerdo a la similitud que tengan hacia la consulta. El resultado contendrá documentos directa e indirectamente relacionados con la consulta. Los documentos directamente relacionados son aquellos documentos que contienen tal cual a los elementos de la consulta. De manera indirecta también se identifican y devuelven los documentos que contienen sinónimos, elementos semánticamente relacionados, hipónimos o hiperónimos de los elementos de la consulta. No obstante, la importancia de los documentos indirectos será proporcional al peso de la relación que tengan los sinónimos, elementos semánticamente relacionados, hipónimos o hiperónimos respecto a los elementos de la consulta.

Para identificar a los documentos directamente relacionados a los elementos de la consulta se calcula una *ponderación directa*. Por otro lado, también se calcula una *ponderación indirecta* para los documentos relacionados indirectamente.

A partir de las ponderaciones *directa* e *indirecta* se calcula una *ponderación total*, que representa la similitud total del documento hacia la consulta.

Para que un documento sea devuelto como parte del resultado de la consulta se consideró que debe tener una ponderación de similitud de al menos 50% hacia la consulta. Este umbral se estableció mediante inspección manual de 250 experimentos de consultas (de acuerdo al Teorema del Límite Central [34]) para conocer la relevancia hacia la consulta de los documentos devueltos. Aquellos documentos que tienen una ponderación menor al umbral establecido tienen poca, muy poca o nula relación con la consulta.

La consulta deseada se transforma en un vector a partir del cual se forman vectores por cada documento que esté relacionado con la consulta, éstos se denominan *vector consulta* y *vector documento* respectivamente.

3.2.1 Vector consulta y vector documento

Una consulta puede formarse por una lista de términos separados por comas que denotan sustantivos, verbos, una relación sustantivo-verbo o una frase. En general, un sustantivo o verbo puede ser *simple* (por ej. *Web* o *learn*) o *compuesto* (por ej. *Semantic Web* o *stand up*). Si se desea buscar una frase, de ésta se extraen los sustantivos y verbos. A cada término de la consulta se le extrae su lexema, que son los que realmente se buscan en el grafo. La consulta se transforma a un *vector consulta*, el cual se representa como indica la Fórmula 1.

$$c = \{ltc_1, ltc_2, ltc_3, \ldots, ltc_n\} \equiv \vec{c} = \{p_{ltc_1}, p_{ltc_2}, p_{ltc_3}, ..., p_{ltc_n}\} \quad (1)$$

donde ltc_i es un lexema del término *i* de la consulta, la cual puede tener hasta *n* términos; $\vec{c}$ es el *vector consulta* donde cada ltc_i tiene un peso p_{ltc_i}, el cual representa el grado de aparición deseado del término por parte del usuario en los resultados de la consulta.

A partir del *vector consulta* $\vec{c}$ se construye un *vector documento* $\vec{d}_j$ para cada documento d_j representado en el grafo, tomando en cuenta cada uno de los elementos y su orden en el *vector consulta*, como puede verse en la Fórmula 2.

$$\vec{d}_j = \{p_{ltc_1}, p_{ltc_2}, p_{ltc_3}, ..., p_{ltc_n}\} \quad (2)$$

donde d_j es un documento en el grafo que estará representado por el peso que tiene cada uno de los elementos del vector consulta dentro del documento. Nótese que el vocabulario que se emplea para representar el documento está formado únicamente a partir de los elementos del vector consulta y no por el vocabulario completo (todos los términos de todos los documentos), lo cual hace manejable la representación vectorial de los documentos en memoria. Así, los *m* documentos del grafo quedarán representados como muestra la Fórmula 3.

$$\begin{aligned}
&\vec{d}_1 = \{p_{ltc_1 d_1}, p_{ltc_2 d_1}, p_{ltc_3 d_1}, ..., p_{ltc_n d_1}\} \\
&\vec{d}_2 = \{p_{ltc_1 d_2}, p_{ltc_2 d_2}, p_{ltc_3 d_2}, ..., p_{ltc_n d_2}\} \\
&\vec{d}_3 = \{p_{ltc_1 d_3}, p_{ltc_2 d_3}, p_{ltc_3 d_3}, ..., p_{ltc_n d_3}\} \quad (3) \\
&\ldots \ldots \ldots \\
&\vec{d}_m = \{p_{ltc_1 d_m}, p_{ltc_2 d_m}, p_{ltc_3 d_m}, ..., p_{ltc_n d_m}\}
\end{aligned}$$

donde $p_{ltc_i d_j}$ representa el peso del lexema del término *i* de la consulta en el documento *j*, para los *m* documentos.

Por ejemplo, supóngase la consulta formada por los siguientes términos "*combine related data on different documents*". Aunque los lexemas de la consulta son *combine*, *relate*, *data*, *different* y *document*, los términos que conforman el vector consulta son *combine*, *relate*, *data* y *document*; *different* no se toma en cuenta pues no es verbo ni sustantivo. De tal manera que el vector consulta queda como *{combine, relate, data, document}*. Dado que los términos del vector consulta son los deseados por el usuario, éstos tienen el máximo peso, por lo que el vector

consulta se traduce a `{1.0, 1.0, 1.0, 1.0}`, que será el vector contra el que se compararán los vectores documento, como se muestra a continuación.

$$c = \{combine, relate, data, document\} \equiv \vec{c} = \{1.0,1.0,1.0,1.0\} \quad (4)$$

Supóngase ahora que los documentos d_1, d_2 y d_3 contienen con diferentes pesos a los elementos de la consulta, los documentos se representan de la forma como se muestra en la Fórmula 5.

$$\begin{aligned} \vec{d}_1 &= \{0.8,1.0,0.7,1.0\} \\ \vec{d}_2 &= \{1.0,0.9,0.7,1.0\} \\ \vec{d}_3 &= \{0.7,0.0,0.8,0.7\} \end{aligned} \quad (5)$$

El vector documento $\vec{d}_1$ indica que el documento d_1 contiene a todos los lexemas deseados de los términos de la consulta, del primer lexema se tiene un hipónimo (0.8), el segundo lexema aparece tal cual en el documento (1.0), del tercer lexema el documento contiene un hiperónimo (0.7) y el cuarto lexema aparece tal cual en el documento (1.0). El vector documento $\vec{d}_2$ indica que el documento d_2 contiene a todos los lexemas de la consulta, el primer lexema aparece en el documento (1.0), del segundo lexema se tiene un término semánticamente relacionado (0.9), del tercer lexema el documento contiene un hiperónimo (0.7) y el cuarto lexema aparece tal cual en el documento (1.0). Algo similar ocurre con el vector documento $\vec{d}_3$, a excepción que no contiene absolutamente nada relacionado con el segundo lexema.

3.2.2 Ponderación directa

Dado que cada documento se modela con base al vector consulta, se obtienen *m* vectores documento y un vector consulta. Para determinar qué tan similares son los documentos respecto a la consulta se emplea la medida de *similitud de coseno* [38], como se muestra en la Fórmula 6.

$$similitud_{\cos} = \cos(c, d_j) = \frac{\vec{c} \cdot \vec{d}_j}{\|\vec{c}\| \cdot \|d_j\|} = \frac{\sum_{i=1}^{n} p_{ltc_i} \cdot p_{ltc_i d_j}}{\sqrt{\sum_{i=1}^{n} (p_{ltc_i})^2} \cdot \sqrt{\sum_{i=1}^{n} (p_{ltc_i d_j})^2}} \quad (6)$$

donde p_{ltc_i} representa el peso de cada lexema de la consulta *c* y $p_{ltc_i d_j}$ representa el peso lexema *i* de la consulta en el documento d_j. Esto es, dados el vector consulta $\vec{c}$ y un vector documento $\vec{d}_j$ la similitud de coseno es un producto escalar y magnitud, donde el numerador representa el producto escalar de los vectores $\vec{c}$ y $\vec{d}_j$, el denominador es el producto de la distancia euclidiana de ambos vectores.

Con esta medida se obtiene el ángulo del coseno que se forma entre el *vector consulta* y un *vector documento*; de manera que cada vector documento se compara con el vector consulta. Cada vector representa una línea en el plano cartesiano, por lo que la medida de similitud de coseno calcula la separación entre la línea que representa el documento y la que representa la consulta. El resultado de la función coseno es un valor entre [0, 1], el cual es 1 cuando no existe separación y menos de 1 cuando el ángulo toma cualquier otro valor.

3.2.3 Ponderación indirecta

Dado que es posible que en el grafo generado se encuentren pares de sustantivos y verbos muy repetidos, la ponderación directa puede dar mayor importancia a los documentos que contienen alta frecuencia de esos términos comunes. Para solventar esta situación se emplea una ponderación indirecta de los documentos mediante la aplicación

del algoritmo *PageRank* en su versión básica (PRB) [39]. El PRB se aplica a un subgrafo correspondiente a los documentos mejor calificados en la ponderación directa. Esto es, primero se identifican los documentos con más altas ponderaciones (arriba del 50% de similitud respecto a la consulta), después se obtienen los sustantivos y verbos que contienen. De éstos se identifican los nodos correspondientes en el grafo y se buscan las relaciones directas entre documentos de acuerdo a lo que indica el nodo (ignorando la transitividad de nodos), tal como muestra la Figura 4. Nótese en la Figura que el nodo oscuro está relacionado, en esta ocasión, con los nodos de color gris, aunque no necesariamente estén directamente conectados a él; por su parte algunos nodos directamente conectados al nodo oscuro no se toman en cuenta. Esto se debe a que los nodos grises pertenecen a documentos altamente calificados en la ponderación directa; el nodo oscuro está relacionado con dichos documentos mediante relaciones semánticas de sus sustantivos.

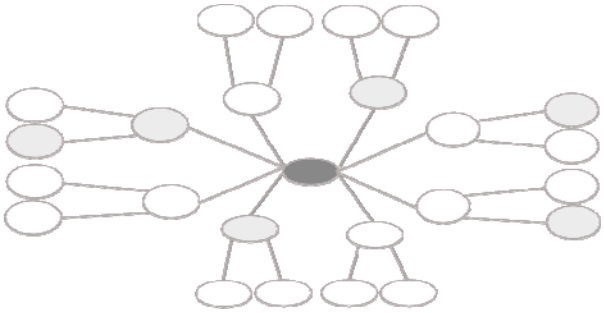

Figura 4. Nodos relacionados en ponderación indirecta

Así, se tiene un subgrafo que denota las relaciones directas entre los documentos mejor calificados en la ponderación directa. De esta manera el PRB puede identificar cuáles son las relaciones más fuertes y las más débiles entre dichos documentos según las referencias que hagan o que les hagan (conexiones entre nodos). La idea es asignar un peso a cada documento del subgrafo de acuerdo a la relación que tenga respecto a la consulta. El algoritmo PRB aplicado se muestra en la Fórmula 7.

$$PRB(D) = (1-c) + c \cdot \sum_{j=1}^{m} \frac{PRB(d_j)}{N(d_j)} \tag{7}$$

donde *PRB(D)* es el valor de *PageRank básico* para el documento *D*, *c* es una constante con valor entre [0,1] (de acuerdo a la literatura se empleó 0.85); *m* representa el total de documentos relacionados con *D* (es decir, los documentos que apuntan hacia *D* o *D* apunta hacia ellos). $PRB(d_j)$ representa el valor de aplicar PRB al documento j-ésimo conectado a D y $N(d_j)$ representa el total de enlaces a los que apunta d_j.

3.2.4 Ponderación total

A partir de la ponderación directa y la ponderación indirecta se obtiene un valor de *ponderación total (PT)*, que representa la calificación que obtienen aquellos documentos que están más relacionados a la consulta pero que además son los que tienen más importancia entre los documentos que están relacionados con dicha consulta. Esta ponderación se obtiene mediante la Fórmula 8.

$$PT(d_j) = \frac{1-(\cos(c,d_j)\cdot PRB(d_j))}{-\ln(\cos(c,d_j)\cdot PRB(d_j))} \tag{8}$$

PT sólo toma en cuenta aquellos documentos d_j que están relacionados a la consulta *c*. El valor asignado a cada uno de estos documentos representa qué tan similar es un documento a la consulta respecto a todos los documentos relacionados directamente a la consulta. El término -*ln* es para efecto de normalizar el valor de PT.

4 EVALUACIÓN

Actualmente, a partir del método propuesto se ha implementado una aplicación en Java. Se han empleado las APIs mencionadas en la sección anterior y además se han implementado clases propias. La implementación se ha realizado y evaluado en un equipo de escritorio con Windows 7 y procesador Intel Quad-core con 4 GB en RAM. Con la implementación se hicieron diversas pruebas con el corpus Reuters-21578, pero para efecto de reportar resultados en este artículo se hicieron pruebas con un conjunto de documentos descargados de la *Biblioteca Digital* de la *ACM*[13]. Se optó por la dicha biblioteca debido a que la organización de las publicaciones corresponde con la clasificación de temas de *Computación* de la ACM, la cual es aceptada internacionalmente por diversas instituciones y organizaciones académicas.

El escenario de evaluación que se plantea simula la situación de un usuario que desea organizar documentos técnico-científicos que ha descargado de la *Biblioteca Digital* de la *ACM*, sobre los cuales posteriormente hará consultas. En este escenario se considera que típicamente un usuario puede descargar manualmente los documentos, por lo que se consideró que el conjunto de documentos fuera de 500 ficheros, de los cuales 400 están en formato PDF y 100 en DOCX. Los ficheros PDF fueron directamente descargados, mientras que los ficheros DOCX fueron creados manualmente a partir del contenido de otros ficheros PDF también descargados de la *Biblioteca Digital* de la *ACM*. Se descargaron 100 documentos por cada uno de los siguientes temas:

1) Semantic Web
2) Linked Data
3) Ontologies
4) Semantic similarity
5) Social Networks

Se buscó que los temas tuvieran alguna relación para que hubiera traslape entre los sustantivos y verbos y, por ende, entre los documentos. Para efecto de comprobación, los documentos fueron organizados en sus respectivos directorios de acuerdo al tema. Esta es la situación que normalmente ocurre cuando un usuario típico desea organizar sus documentos: agrupa de manera manual aquellos documentos que considera son similares entre sí, aun así sean del mismo dominio, en el caso del ejemplo *Computación.*

Se generaron 3 grupos de consultas de acuerdo a la longitud de la consulta: 15 consultas cortas, 15 consultas medias y 15 consultas largas; en total 45 consultas. En las Tablas 1, 2 y 3 se muestran algunos ejemplos de las consultas por grupo.

Tabla 1. Ejemplos de consultas de longitud corta

Profile
Integration
Mapping data
Ontology learning
Merging information

[13] *ACM International Conference Proceeding Series*

Tabla 2. Ejemplos de consultas de longitud media

Semantic web application
Exploring the hidden web
Retrieving relevant information
Integrating content semantics
Interoperability of related data

Tabla 3. Ejemplos de consultas de longitud larga

The extraction of tuples from relational databases
Data management on semantic graphs from the Web
Analysis of annotation propagation in social networks
The use of social data for recommender systems
Software development by using formal ontologies

La implementación (denominada *PROPUESTA*) se comparó contra dos herramientas disponibles en la Web: Google Desktop[14] y LogicalDoc[15]. Si bien tanto Google Desktop como LogicalDoc ofrecen diversas funcionalidades para el usuario, en la evaluación sólo se tomaron en cuenta las funcionalidades de organización y búsqueda de documentos.

Dado el escenario planteado para la evaluación, se consideró que a un usuario típico, que guarda sus documentos y luego los busca, le interesará recuperar aquellos documentos que son relevantes a las consultas que realice, por lo que el desempeño de la implementación realizada se evalúa mediante las medidas de Recuperación de Información precisión (P), cobertura (C) y medida F1 [40]. Al ser un ambiente controlado se conoce de antemano a qué tema pertenecen los documentos y la relación que pueda existir entre documentos, por lo que se puede saber cuándo un documento es relevante o no hacia la consulta mediante inspección manual. La comparativa de los resultados de las 45 consultas puede verse en las Figuras 5, 6 y 7 respectivamente para precisión, cobertura y medida F1.

[14] download.cnet.com/Google-Desktop [visitado en agosto 2015]. Actualmente Google ya no le da soporte, pero la herramienta continúa siendo funcional.
[15] www.logicaldoc.com [visitado en agosto 2015]

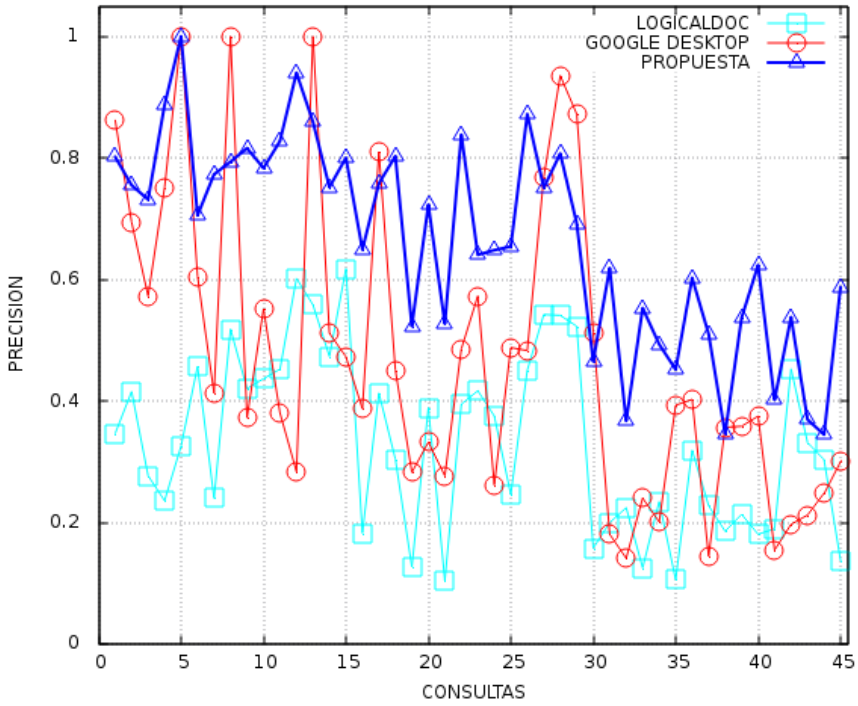

Figura 5. Gráfica comparativa de resultados de *precisión*. En general la precisión obtenida por la PROPUESTA fue mejor en la mayoría de las consultas. Google Desktop tuvo mejores valores en pocas consultas, mientras que LogicalDoc tuvo los valores menores para todas las consultas.

Como puede verse en la Figura 5, en general el desempeño de la aplicación desarrollada (*PROPUESTA*) mostró mejores resultados en cuanto a *precisión*, la variabilidad de los valores obtenidos es poca en comparación con los obtenidos para Google Desktop y LogicalDoc. No obstante, Google Desktop tuvo mejores resultados para las consultas 1, 8, 13, 17, 27, 28, 29 y 30; lo que significa que tuvo el mejor comportamiento en un 17.77% de las consultas, mientras que la PROPUESTA tuvo el mejor comportamiento en el 82.22% de las consultas. LogicalDoc no superó a la PROPUESTA, todos los valores de precisión que obtuvo estuvieron por debajo de los obtenidos por la PROPUESTA.

Considerando el desempeño general de las herramientas comparadas en cuanto a precisión, puede verse que la PROPUESTA tuvo un comportamiento casi homogéneo; como era de esperarse, la PROPUESTA con las consultas cortas tuvo mejores valores de precisión, los cuales bajaron en las consultas de longitud media y bajaron aún más en las consultas de longitud larga. Google Desktop tuvo sobresaltos en su comportamiento tanto en las consultas de longitud corta como en las de longitud media, con valores bajos para las consultas de longitud larga. Si bien LogicalDoc tuvo también un comportamiento casi homogéneo, sus valores fueron siempre los más bajos.

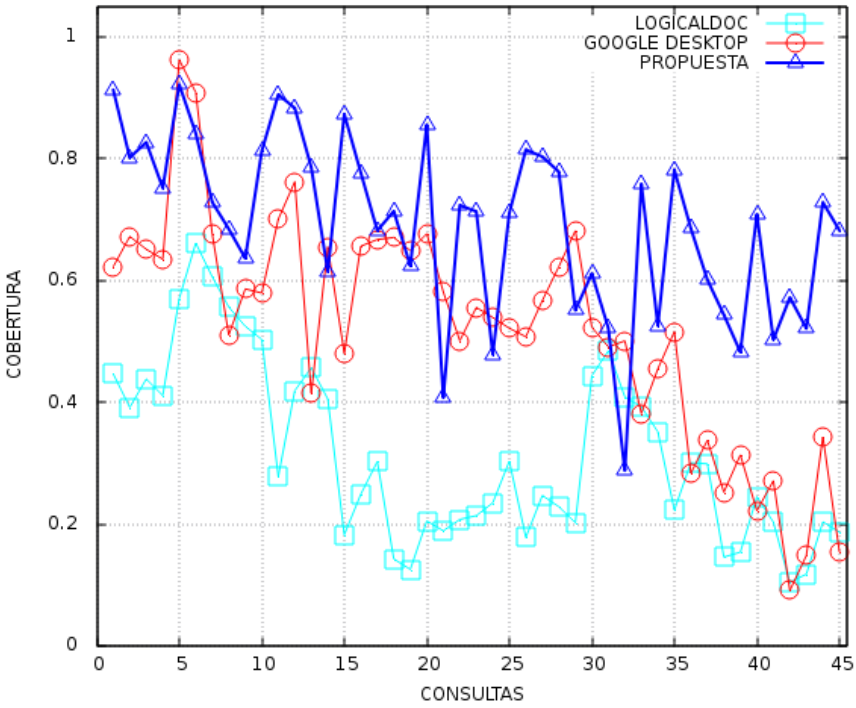

Figura 6. Gráfica comparativa de resultados de cobertura. Aunque la cobertura obtenida por la PROPUESTA tuvo sobresaltos, para la mayoría de las consultas la PROPUESTA obtuvo los mejores valores. Google Desktop tuvo casi un comportamiento uniforme descendente pero sus valores no superaron a la PROPUESTA. LogicalDoc obtuvo los valores más bajos.

Como muestra la Figura 6, la PROPUESTA tuvo los mejores valores para la *cobertura*, los cuales, en general, oscilaron entre 0.4 y 0.9. En general, los valores de Google Desktop oscilaron entre 0.15 y 0.78. Los valores de LogicalDoc oscilaron entre 0.15 y 0.5. Los valores de la PROPUESTA fueron casi homogéneos para todas las consultas, es decir, no hubo sobresaltos que destacaran. Si bien los valores de Google Desktop al principio fueron altos, éstos fueron decreciendo conforme las consultas iban aumentando su longitud, teniendo un descenso con las consultas largas. En cinco consultas Google Desktop tuvo mejores valores que la PROPUESTA. Hubo traslape en los valores de Google Desktop y la PROPUESTA para las consultas cortas y de longitud media. LogicalDoc tuvo los valores más bajos, aunque en las consultas cortas y largas tuvo traslape con los valores de Google Desktop, sólo en una consulta superó a la PROPUESTA.

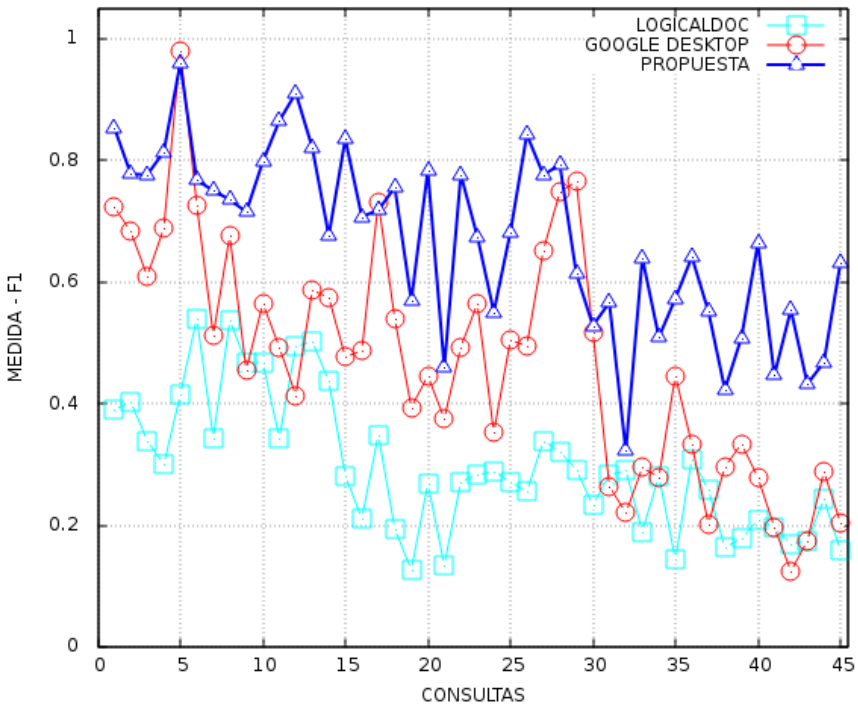

Figura 7. Gráfica comparativa de resultados de medida F1. Aunque la PROPUESTA tuvo sobresaltos, su comportamiento fue, en general, mejor que Google Desktop y LogicalDoc. Google Desktop tuvo sobresaltos muy pronunciados, en las consultas largas obtuvo valores similares a LogicalDoc. LogicalDoc obtuvo casi un comportamiento uniforme descendente, siempre obtuvo los valores más bajos.

En la Figura 7 se muestran los valores de la medida F1, la cual es la media de los valores de precisión y cobertura; con F1 ambas tienen la misma importancia. De esta forma se sintetiza el comportamiento de las herramientas evaluadas. A excepción de dos consultas, la *PROPUESTA* tuvo los valores más altos. Como era de esperarse, se puede ver que su comportamiento fue decayendo de acuerdo a la longitud de las consultas. Google Desktop tuvo valores menores que la PROPUESTA pero mayores que LogicalDoc, su desempeño también fue decayendo conforme la longitud de las consultas, aunque tuvo tres sobresaltos muy pronunciados. En cuatro consultas tuvo valores similares a la PROPUESTA y en dos la superó, pero en el resto la PROPUESTA tuvo mejores valores. LogicalDoc tuvo los valores más bajos, en las consultas medias y largas tuvo un desempeño casi similar en ambos casos. En las consultas cortas y largas tuvo algunos valores traslapados respecto a los de Google Desktop, pero no lo superó en el resto de las consultas. No hubo traslape en los valores de LogicalDoc y la PROPUESTA.

Aunque Google Desktop y LogicalDoc son sistemas cerrados y no se pudo conocer el funcionamiento interno de estas herramientas, dadas sus características públicas y funcionalidades de organización y búsqueda, consideramos que la PROPUESTA tiene las siguientes ventajas respecto a estas herramientas:

- No se emplean metadatos para organizar los documentos
- No se emplean todos los términos de un documento para representarlo
- No se modelan los documentos, sino los temas de los documentos mediante el grafo de sustantivos-verbos
- La búsqueda se centra en los temas, consecuentemente se devuelven los documentos que contienen tales temas
- Se emplean relaciones semánticas (sinónimos, términos semánticamente relacionados, hipónimos e hiperónimos)

Como puede verse en las Figuras 5,6 y 7, el desempeño de los sistemas evaluados tuvo sobresaltos debido a que en algunas consultas se tuvo éxito y en otras no tanto, por lo que el valor de la media no sería representativo del desempeño general de los sistemas. Es así como se optó por obtener la mediana de los valores como una medida más robusta del desempeño, estos valores se muestran en la Tabla 7 y se ilustran en la Figura 8. Consideramos que los resultados de la PROPUESTA son mejores en la comparativa debido a que se emplea una extensión del MEV para la representación de documentos. En lugar de representar los términos se representan relaciones semánticas relevantes de un documento, estas relaciones se plasman en un grafo. Lo anterior permite relacionar directamente (sin necesidad de cálculos adicionales) temas de un mismo documento y entre diferentes documentos, algo que no se puede hacer con el MEV básico.

Tabla 7. Comparación de valores de mediana de *precisión, cobertura y medida F1*

MEDIDA	PROPUESTA	GOOGLE DESKTOP	LOGICALDOC
Precisión	0.6922	0.4025	0.3256
Cobertura	0.7128	0.5387	0.2777
Medida F1	0.6811	0.4869	0.2823

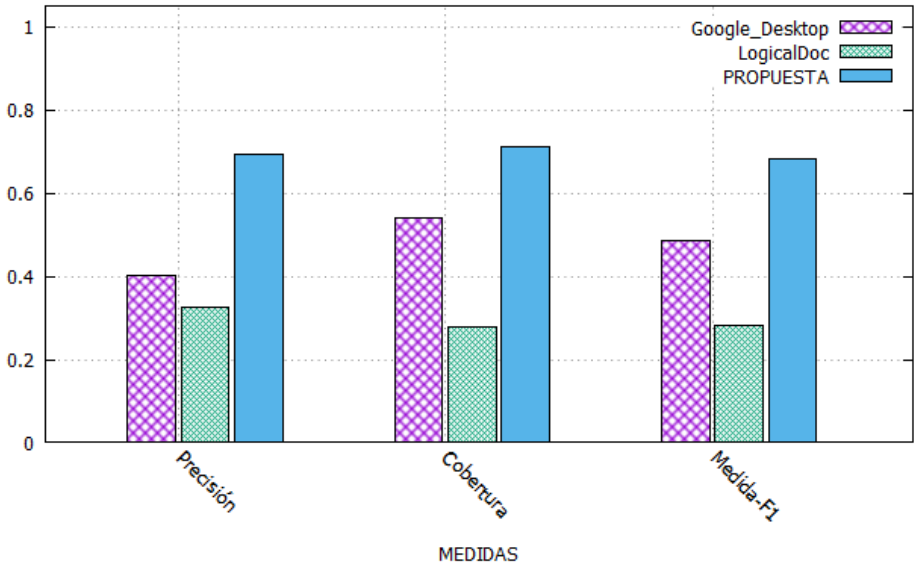

Figura 8. Gráfica comparativa de los valores de mediana de *precisión, cobertura y medida F1*

Ahora bien, aunque el comportamiento de la PROPUESTA tanto en la *precisión, cobertura* como en la medida F1, en general, fue bueno, no fue así en la etapa de indexación, donde tuvo los tiempos más prolongados para indexar los 500 documentos. Como puede verse en la Tabla 8, el mejor tiempo lo tuvo Google Desktop seguido por LogicalDoc.

Tabla 8. Tiempos en el proceso de indexación

Herramienta	Tiempo (segundos)
Google Desktop	92
LogicalDoc	116
PROPUESTA	284

La PROPUESTA tardó más del doble que LogicalDoc y más del triple que Google Desktop. Consideramos que esto se debe a que el algoritmo de agrupamiento (CBC) hace uso intensivo de la memoria para identificar las clases semánticas entre sinónimos, términos semánticamente relacionados, hipónimos e hiperónimos. Algo que también debe tomarse en cuenta es que el código de la PROPUESTA no ha sido optimizado, lo cual es muy probable que no sea el caso para los códigos de Google Desktop y LogicalDoc.

5 Conclusiones

Hoy en día hay una creciente necesidad de guardar información en documentos (correos, noticias, apuntes, notificaciones, etc.). Al mismo tiempo la necesidad de acceder a esos documentos incrementa; dicho acceso se complica cuando la cantidad de documentos es grande. Si bien el usuario puede seguir algún esquema de organización (temas, fechas, autor, etc.), al pasar el tiempo tales esquemas dejan de ser funcionales.

En este artículo se describe un método para organizar documentos escritos en Inglés haciendo uso de una representación basada en grafos. Cada documento se representa mediante un grafo, mediante la identificación de los temas de los documentos se genera un grafo más grande que modela a todos los documentos. En dichos grafos los nodos representan los sustantivos relevantes que contienen los documentos. De esta manera se puede acceder a un nodo y a partir de las relaciones con otros nodos (sinónimo, término semánticamente relacionado, hipónimo o hiperónimo) se puede llegar a los sustantivos y verbos a los que está relacionado dicho nodo. Un nodo puede estar asociado a más de un documento de tal manera que los documentos están asociados de acuerdo al tema del que trata según sus sustantivos.

El método se implementó mediante una aplicación Java, la cual fue comparada contra dos herramientas con tareas similares para la organización y búsqueda de documentos: Google Desktop y LogicalDoc. Según la experimentación llevada a cabo, en general, el método propuesto tuvo mejor desempeño que las otras herramientas tanto para la *precisión*, *cobertura* y *medida F1*. No obstante, la herramienta tuvo los tiempos más prolongados durante la etapa de indexación de documentos. Si bien el método propuesto tarda en indexar los documentos, los resultados que obtiene al realizar búsqueda son mejores que las otras dos herramientas, por lo que un usuario debería sopesar si desea buenos resultados en sus consultas sin tomar importancia a los tiempos de indexación. Lo cual podría resolverse ejecutando la indexación en los tiempos muertos del equipo donde se instale.

Consideramos como desventaja real el hecho de que el método no puede procesar bien los términos multipalabra. Cuando una consulta contiene alguna multipalabra lo que hace es descomponer el término en los términos que contenga y a partir de ahí hacer la búsqueda. Otra desventaja importante es que si los documentos del grupo de documentos pertenecen a temas sin relación o definitivamente son de dominios diferentes, el método no funciona adecuadamente pues intentará de cualquier forma relacionar los documentos, aunque dicha relación sea mínima o nula; se tendrían que hacer diversas adecuaciones para que funcionara adecuadamente en un escenario como éste.

Referencias

[1] Pamela Ravasio, Sissel Guttormsen Schar, and Helmut Krueger. In pursuit of desktop evolution: User problems and practices with modern desktop systems. *ACM Trans. Comput.-Hum. Interact.*, 11(2):156–180, 2004.

[2] Susan L. Price, Marianne Lykke Nielsen, Lois M.L. Delcambre, Peter Vedsted, and Jeremy Steinhauer. Using semantic components to search for domain-specific documents: An evaluation from the system perspective and the user perspective. *Information Systems* 34:724–752, 2009.

[3] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. *In: 10th International Conference on Conceptual Structures. Conceptual Structures: Integration and Interfaces.* LNCS Vol 2393, pp. 92-106. Springer 2002.

[4] Fausto Giunchiglia, Uladzimir Kharkevich, and Ilya Zaihrayeu. Concept Search. *In: 6th European Semantic Web Conference*, pp. 429–444. Heraklion, Crete, Greece. LNCS 5554 Springer 2009.

[5] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1997.

[6] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, A. Saarela, Self organization of a massive document collection. *IEEE Trans. Neural Networks* 11(3):574–585. 2000.

[7] H. Yin. The Self-Organizing Maps: Background, Theories, Extensions and Applications. *Studies in Computational Intelligence: A compendium*. pp. 715–762, Springer 2008.
[8] Yen, G.G., Wu, Z.: Ranked Centroid Projection: A Data Visualization Approach with Self-Organizing Maps. *IEEE Transaction on Neural Networks* 19(2):245–259, 2008.
[9] Yang, Hsin-Chang, Chung-Hong Lee and Ding-Weng Chen. A method for multilingual text mining and retrieval using growing hierarchical self-organizing maps. *Journal of Information Science* 35(1):3-23, 2009.
[10] Feng, Z., Bao, J., Shen, J.: Dynamic and Adaptive Self Organizing Maps applied to High Dimensional Large Scale Text Clustering. *In: IEEE International Conference on Software Engineering and Service Sciences*, pp. 348–351, 2010.
[11] Pramod Kumar Singh, Mahesh Machavolu, Kusum Bharti, and Ranjith Suda. Analysis of Text Cluster Visualization in Emergent Self Organizing Maps Using Unigrams and Its Variations after Introducing Bigrams. *In: International Conference on SocProS 2011*, pp. 967–978. Springer 2012.
[12] Renato Fernandes Corre and Teresa Bernarda Ludermir. Improving self-organization of document collections by semantic mapping. *Neurocomputing* 70:62–69 2006.
[13] Mohamed Salah Hamdi. SOMSE: A semantic map based meta-search engine for the purpose of web information customization. *Applied Soft Computing* 11:1310–1321. 2011
[14] Lin Guo, Feng Shao, Chavdar Botev, and Jayavel Shanmugasundaram. XRANK: Ranked Keyword Search over XML Documents. *In: 2003 ACM SIGMOD international conference on management of data*, pp. 16-27. ACM New York, NY, USA 2003.
[15] R. Schenkel, A. Theobald, and G. Weikum. Semantic Similarity Search on Semistructured Data with the XXL Search Engine. *Inf. Retr.*, 8(4):521–545, 2005.
[16] Sara Cohen, Jonathan Mamou, Yaron Kanza, Yehoshua Sagiv. XSEarch: A Semantic Search Engine for XML. *In: 29th International Conference on Very Large Data Bases*, pp 45-56, Berlin, Germany. Morgan Kaufmann 2003.
[17] Zografoula Vagena and Mirella M. Moro. Semantic Search over XML Document Streams. *In: Third International Workshop on Database Technologies for Handling XML Information on the Web*. Nantes, France 2008.
[18] Chikashi Nobata, Yutaka Sasaki, Naoaki Okazaki, C.J. Rupp, Jun'ichi Tsujii, and Sophia Ananiadou. Semantic Search on Digital Document Repositories based on Text Mining Results. *In: International Conference on Digital Libraries and the Semantic Web* pp. 34-48. Trento, Italia, 2009.
[19] Anália Lourenco, Rafael Carreira, Daniel Glez-Peña, José R. Méndez, Sónia Carneiro, Luis M. Rocha, Fernando Díaz, Eugénio C. Ferreira, Isabel Rocha, Florentino Fdez-Riverola, and Miguel Rocha. BioDR: Semantic indexing networks for biomedical document retrieval. *Expert Systems with Applications* 37:3444–3453. 2010.
[20] Jinoh Oh, Taehoon Kim, Sun Park, Hwanjo Yu, and Young Ho Lee. Efficient semantic network construction with application to PubMed search. *Knowledge-Based Systems* 39:185–193. 2013.
[21] Henrik Eriksson. The semantic-document approach to combining documents and ontologies. *Int. J. Human-Computer Studies* 65:624–639. Elsevier 2007.
[22] Eduardo Lupiani-Ruiz, Ignacio García-Manotas, Rafael Valencia-García, Francisco García-Sánchez, Dagoberto Castellanos-Nieves, Jesualdo Tomás Fernández-Breis, and Juan Bosco Camón-Herrero. Financial news semantic search engine. *Expert Systems with Applications* 38:15565–15572. 2011
[23] Xutang Zhang, Xin Hou, Xiaofeng Chen, and Ting Zhuang. Ontology-based semantic retrieval for engineering domain knowledge. *Neurocomputing* 116:382–391, 2013.
[24] Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi and Daniela Petrelli. Hybrid Search: Effectively Combining Keywords and Semantic Searches. *In: 5th European semantic web conference on the semantic web: research and applications* pp. 554-568. Tenerife, Spain. Springer 2008.
[25] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. Indexing with WordNet synsets can improve text retrieval. Proceedings of the COLING/ACL'98 Workshop on Usage of WordNet for NLP, pp 38-44. Montreal, 1998.
[26] D.R. Recupero, A new unsupervised method for document clustering by using WordNet lexical and conceptual relations, *Information Retrieval* 10(6): 563–579, Springer, 2007.
[27] Chun-Ling Chen, Frank S.C. Tseng, Tyne Liang. An integration of WordNet and fuzzy association rule mining for multi-label document clustering. *Data & Knowledge Engineering* 69:1208–1226, Elsevier, 2010.

[28] Mauro Dragoni, Célia da Costa Pereira, Andrea G.B. Tettamanzi . A conceptual representation of documents and queries for information retrieval systems by using light ontologies. *Expert Systems with Applications* 39(12): 10376-10388, Elsevier, 2012.
[29] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science (JASIS), 41 (6), 391{407.
[30] Turney, P.D.: The latent relation mapping engine: algorithm and experiments. J. Artif. Intell. Res. 33(1), 615{655 (2008)
[31] Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18 (11), 613{620.
[32] Harris, Z. (1954). Distributional structure. Word, 10 (23), 146-162.
[33] Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier de Lacalle, and Eneko Agirre. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. *In: 20th ACM International Conference on Information and Knowledge Management* pp. 2317–2320, New York, NY, USA, 2011.
[34] John Rice. Mathematical Statistics and Data Analysis, Duxbury Press 1995.
[35] Robert M. Losee. The Effect of Assigning a Metadata or Indexing Term on Document Ordering. *Journal of the American Society for Information Science and Technology* 64(11):2191–2200, 2013.
[36] Patrick Pantel. Clustering by committee. *Ph.D. thesis*, Department of Computing Science. University of Alberta. 2003.
[37] Lin, D. and Pantel, P. 2001a. Induction of semantic classes from natural language text. In Proceedings of SIGKDD-01. pp. 317–322. San Francisco, CA.
[38] Michael E. Lesk. Word-word associations in document retrieval systems. *American Documentation* 20(1):27–38, 1969.
[39] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30:107–117. 1998
[40] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *An Introduction to Information Retrieval*. Cambridge University Press. 2008.