



Sistema de recuperación de información apoyado en Web Semántica

Document search system supported by semantic web

JARAMILLO-VALBUENA, Sonia 1; DUQUE- ARANGO, John J. 2; PULGARÍN-GIRALDO, Róbinson 3

Recibido: 09/10/2019 • Aprobado: 20/02/2020 • Publicado 30/04/2020

Contenido

- [1. Introducción](#)
- [2. Estado del arte](#)
- [3. Metodología](#)
- [4. Resultados](#)
- [5. Conclusiones](#)

[Referencias bibliográficas](#)

RESUMEN:

Un sistema de recuperación de información permite presentar información relevante al usuario dando respuesta a una consulta. Diferentes elementos pueden ser seleccionados luego de la misma, siendo un ranking el que determina cual tiene mayor correspondencia con el dominio de búsqueda. El objetivo de este artículo es presentar un sistema que se apoya en técnicas de recuperación de la información y Web Semántica. Esta aproximación usa archivos anotados semánticamente, a partir de los cuales se construye un índice invertido con los términos ontológicos del corpus. Cada elemento del índice es un conjunto de términos ontológicos que apunta a los documentos asociados semánticamente. Para procesar una búsqueda, el sistema intenta asociar el query con uno o más términos de la ontología, dando lugar a un vector TF de términos ontológicos. Si esto no es posible, se obtiene la representación TF del documento. Finalmente se efectúa un proceso de emparejamiento entre la representación vectorial del query y los documentos del corpus, que permite seleccionar los documentos que deben ser mostrados al usuario. El sistema fue implementado en Eclipse y fue probado con un dataset del campo biomédico. Sin embargo, la metodología utilizada puede emplearse en cualquier otro contexto. Los resultados experimentales muestran la eficacia del mismo, al presentar al usuario una mayor correspondencia de los resultados con el dominio de búsqueda.

Palabras clave: Algoritmo Apriori, ontología, emparejamiento, web semántica

ABSTRACT:

An information retrieval system presents relevant information to the user in response to a query. This kind of system selects one or more elements to answer the query and uses a ranking to determine which one has greater correspondence with the search domain. In this paper, we present a document retrieval system that uses Semantic Web. This approach uses semantically annotated files to generate an inverse index with the ontological terms of the body. Each element of the index is a set of ontological terms that points to semantically associated documents. The system, to perform a search, attempts to associate the query with one or more terms of the ontology, resulting in a TF vector of ontological terms. If this is not possible, obtains the TF representation of the document. Finally, the system performs a matching process between the vector representation of the query and the documents of the corpus, which allows selecting the documents that should be shown to the user. We have implemented our approach on Eclipse. We tested the prototype with a biomedical dataset. However, the methodology used can be used in any other context. The experimental results show its effectiveness, by presenting the user with a greater correspondence of the results with the search domain.

Keywords: Apriori algorithm, ontology, matching, semantic web

1. Introducción

La era del Big data ha traído consigo diversos desafíos. Mejorar los resultados de las búsquedas para lograr que los documentos encontrados tengan mayor relevancia, contexto y significado para el usuario, es uno de ellos. Esto puede conseguirse mediante el uso de Web semántica, al añadir a los documentos etiquetas XML, para dotar de semántica a los archivos de texto. Tales etiquetas se asocian con modelos semánticos denominados ontologías. Una ontología es una especificación formal de una conceptualización compartida (Borst, 1997; Jaramillo & Londoño, 2014), que permite modelar individuos, propiedades y conceptos. A partir de una ontología se puede construir una base de conocimiento, dado que la misma describe un dominio (Perinan-Pascual & Arcas-Tunez, 2014). Una ontología, es una estructura jerárquica, que puede verse como vocabulario común referente a una determinada área. Las ontologías son una forma de representación, que permiten intercambio de información y además, facilitan el proceso de inferencia de conocimiento (Yuniel & Alianis, 2012).

En la actualidad, la Web Semántica se consolida como una infraestructura apoyada en metadatos que permiten razonar, ampliando las capacidades de los sistemas de búsqueda, mediante el uso de relaciones tales como sinonimia, parentesco, entre otras, definidas en la ontología (Jaramillo & Londoño, 2014).

Para que la infraestructura sea global, la web semántica se apoya en elementos tales como XML, XML Schema, RDF, SPARQL y OWL. El Lenguaje de Marcado Extensible, XML, es un lenguaje de marcado que permite estructurar contenidos al interior de los archivos. El XML Schema es un lenguaje para restringir y fijar la estructura al interior de los documentos XML. Por su parte RDF, presenta de forma descriptiva los recursos disponibles en la página web, fotos. SPARQL se apoya en diferentes fuentes de datos para la realización de consultas sobre los recursos de la web semántica, usando el lenguaje RDF. Finalmente, OWL permite la creación de vocabularios específicos que puedan relacionarse con otros recursos, logrando con ello describir un área específica de conocimiento (Perez, 2016).

Si bien, hoy en día existen diversas formas de acceder a la información disponible en un corpus, aún son latentes 2 importantes problemáticas para la recuperación de información. La primera de ellas, es la posibilidad de altas tasas de falso positivos, es decir, de resultados que no tienen significado para el usuario. La segunda, se relaciona con la detección de elementos que deberían ser mostrados como resultado de la búsqueda (falsos negativos). Una propuesta para abordar esta problemática es apoyarse en ontologías para que los resultados presentados al usuario tengan mayor correspondencia con el dominio de búsqueda (Jaramillo & Londoño, 2014).

El objetivo del trabajo es mostrar un nuevo motor, que proporciona documentos con mayor cercanía en relación al *query*. La estrategia descrita emplea documentos etiquetados con una ontología, a partir de los cuales se elabora un índice invertido con los términos ontológicos presentes en el corpus. Para procesar el query, el sistema trata de asociarlo con uno o más términos de la ontología, generando un vector TF de términos ontológicos. Si esto no es posible, se construye una representación TF del documento. Finalmente se efectúa un proceso de matching entre la representación vectorial del query y la representación vectorial del corpus, esto permite identificar los documentos que deben ser mostrados al usuario. El proceso de matching se hace usando similitud de coseno. Esta es una medida de similitud que mide el ángulo entre dos vectores en el espacio vectorial. La distancia angular se considera una métrica de la similitud entre documentos (Jaramillo & Londoño, 2014).

Este artículo se estructura de la siguiente forma: En la sección 2 se presentan los trabajos relacionados. En la sección 3 se describe el modelo propuesto. En la sección 4 se dan detalles de implementación y se describe la evaluación que se lleva a cabo. Finalmente, en la sección 5 se dan las conclusiones.

2. Estado del arte

Existen varios estudios que hacen uso de ontologías para mejorar los resultados de las búsquedas. Generalmente, tales sistemas las utilizan para extender el query, mediante el uso de relaciones tales como sinonimia y parentesco.

En Jiang & Tan (2006), se describe OntoSearch, un sistema que usa documentos anotados semánticamente para generar una red semántica mediante la aplicación de Difusión de activación (Anderson, 1983). Esta técnica permite inferir la importancia de los conceptos ontológicos dentro del dominio respecto a una petición de información dada por el usuario y el uso de conceptos

semilla. Luego de que usuario ingresa una petición de información, el sistema obtiene una lista de documentos que tienen los términos ingresados, y también, los conceptos ontológicos relacionados. Los términos ontológicos encontrados en el corpus son usados como elementos semilla para generar la red semántica. Mediante la aplicación de una aproximación heurística se determina el grado de fuerza de las relaciones semánticas con respecto al dominio ontológico. Finalmente se calcula el ranking para mostrar los documentos, mediante el uso de la medida de similitud de coseno (Jaramillo & Londoño, 2014).

ORank, presentada en (Shamsfard et al., 2006), es otra aproximación que se apoya en un modelo conceptual ontológico. ORank recibe documentos, que anota con etiquetas semánticas. Luego mediante la aplicación de técnicas estadísticas y conceptuales construye un conjunto de vectores de documentos referente al corpus, cada uno contiene etiquetas y pesos. Luego de que el usuario hace la petición de información se inicia un proceso de extracción de frases, para identificar si las mismas están en la ontología o en las etiquetas de los documentos. Si esto ocurre, ambas palabras y frases se envían como argumento al algoritmo de difusión de activación. Los pesos más altos se asignan a frases. Para mejorar los resultados de la búsqueda del query se expande mediante WordNet (Fellbaum, 2005) logrando con ello la posibilidad de utilizar, entre otros, hipónimos, significados y sinónimos. WordNet es una base de datos léxica que agrupa palabras en idioma inglés. ORank recorre la ontología, para obtener los conceptos ontológicos relacionados con el query y los valores de activación. Finalmente, se convierte el query en un vector que se compara con los vectores de los documentos. Lo anterior, permite identificar la relevancia de con el dominio de búsqueda (Jaramillo & Londoño, 2014).

En Zou (2008), se hace referencia a una metodología basada en ontologías para búsqueda expandida semánticamente, que emplea una ontología de dominio, un algoritmo para anotación semántica y un algoritmo que permite la expansión semántica (Jaramillo & Londoño, 2014).

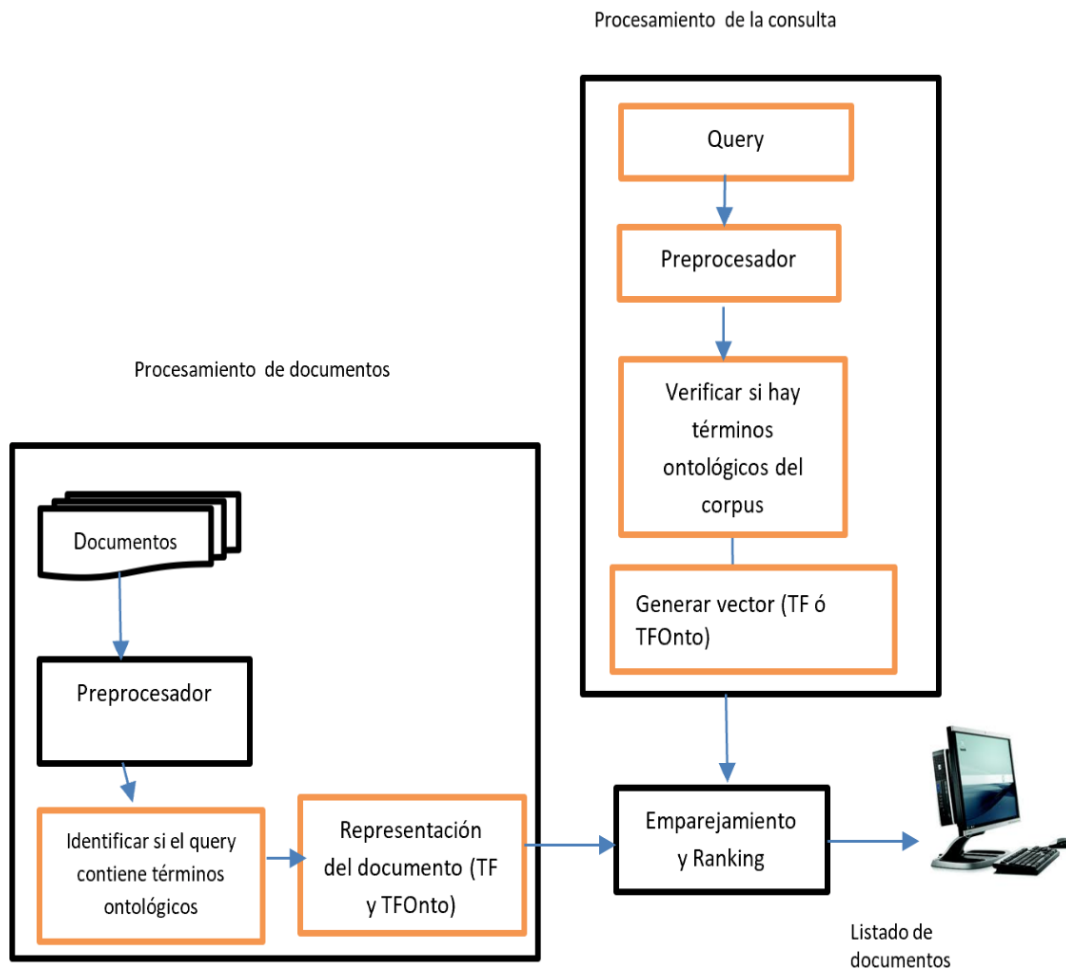
La aproximación presentada en (Wang & Tsai, 2008) describe un sistema que permite recuperar objetos de aprendizaje. Hace uso de 2 importantes características: un mecanismo para eliminación de ambigüedad y un algoritmo para expandir la petición de información apoyada en ontologías. Para dar una respuesta al usuario, el sistema propuesto efectúa una etapa de pre-procesamiento, que consiste en la eliminación de las palabras vacías y aplicación de técnicas de lematización. Luego, con los conceptos ontológicos hallados en el query se construye uno o más árboles de intención del usuario, este árbol es un subárbol dentro de la jerarquía ontológica. Posteriormente, se calcula por cada uno de los árboles un puntaje referente a ambigüedad, este puntaje permite seleccionar el mejor de los árboles. A continuación, se expanden los términos del query, para lo cual se toman en consideración correlaciones semánticas y halla la distancia semántica, para finalmente listar los objetos de aprendizaje que responden al query.

Finalmente, en (Jaramillo & Londoño, 2014) se presenta un sistema que usa Hadoop para generar un índice que permite buscar documentos. Se generan lattices tanto para el corpus como para el query, luego de lo cual se realiza un proceso de matching.

3. Metodología

El sistema propuesto tiene 3 módulos, a saber: procesamiento de documentos (PD), el módulo de procesamiento del query (PQ) y el módulo de emparejamiento-ranking (ER), ver Figura 1.

Figura 1
Arquitectura general del sistema propuesto
basada en (Jaramillo & Londoño, 2014)



Fuente: Imagen propia

Módulo PD se apoya en 2 componentes: ontologías y un corpus que está anotado con la ontología CL, de la forma: `<term sem="CL:idDelTermino">termino</term>`. En este caso, la etiqueta `<term sem="CL: 0000008"> craneal neural crest cell</term>` esto quiere decir que de la ontología CL se tiene el término con código 0000000 y un label "craneal neural crest cell". La etiqueta puede tener 1 o más palabras.

El Módulo PD tiene 3 objetivos. El primero de ellos es obtener la representación vectorial de los documentos, para ello se toman todas las palabras encontradas en el documento, se eliminan las palabras vacías y se obtienen las raíces de los términos (*stemming*). Con ellos, se calculan las frecuencias crudas y se genera un vector TFClásico del documento. El segundo, es extraer los términos ontológicos con sus correspondientes frecuencias. El último objetivo, es construir un índice invertido a partir de los términos ontológicos hallados, y combinaciones de ellos. Lo anterior permite genera el vector TFOnto del corpus (ver Figura 2).

Figura 2

- a) Términos ontológicos hallados en el corpus
- b) Estructura del índice invertido generado

doc1, S_00	Archivo de salida S_00, doc1,doc2,doc3 S_001, doc1,doc2 S_002, doc4 P_000, doc3, doc4 S_00 S_001, doc1, doc2, doc3 S_00 P_00, doc3 S_01 P_00, doc3 S_002 P_00, doc4 S_00 S_001 P_00, doc3
doc1, S_001	
doc2, S_00	
doc2, S_001	
doc3, S_00	
doc3, P_00	
doc4, S_002	
doc4, P_00	
doc1, S_00 S_001	
doc2, S_00 S_001	
doc3, S_00 P_00	
doc4, S_002 P_00	

Fuente: Imagen propia

Módulo PQ. Se apoya en 2 componentes: ontologías y la petición de información ingresada por el usuario. Del query se eliminan los *stopwords*. Luego se intentan asociar los términos ingresados por el usuario con uno o más términos ontológicos. Las combinaciones de términos se obtienen usando la aproximación presentada en (Agrawal & Srikant, 1995) y (Zaki, 2000), logrando de esta forma generar combinaciones frecuentes de términos. Solamente se conservan las combinaciones que tengan documentos relacionados semánticamente (Ceglar & Roddick, 2006; Jaramillo & Londoño, 2014). Por ejemplo, el usuario ingresa las siguientes palabras: "primary cell pigment line cell visual" y el procesador devuelve que cell está asociado con S_00, primary cell line cell con S_01, pigment cell con S_002 y visual cell con S_002. Si se identifica que alguno de los términos ingresados por el usuario corresponde a un sinónimo el sistema debe retornar el término ontológico oficial equivalente al mismo. Con los términos ontológicos hallados, y las combinaciones de ellos, se construye un vector TF_{Onto} para el query, en el que consideran únicamente términos ontológicos. Si no es posible obtener ningún término ontológico a partir del query, se construye un vector TF_{Clásico} con los términos obtenidos después de aplicar stemming y su respectiva frecuencia.

Módulo ER. Este módulo se encarga de hallar la similitud de coseno entre el vector del query y los diferentes vectores del corpus. Si hay vector TF_{Onto} para el query, se halla la similitud con los vectores TF_{Onto} del corpus. De lo contrario, se halla la similitud entre el vector TF_{Clásico} del query y TF_{Clásico} de los documentos del corpus. El sistema de búsqueda siempre da prioridad a encontrar términos ontológicos por encima de términos normales, por ello se verifica siempre, en primer lugar, la existencia de TF_{Onto} para el query.

El pseudocódigo usado para el Sistema de recuperación de información se presenta en las Figuras 1-3. En primer lugar, la función *emparejarVectoresTF*, mostrada en la Figura 3, calcula la similitud de coseno, tal como se presenta en la Ecuación 1. El método devuelve valores entre cero y uno, donde 1 representa mayor cercanía entre el query y el documento. Para este método, vectorTF es el vector correspondiente a cada documento y vectorTFQ es el vector del query.

Figura 3
Función emparejarVectoresTF

```

Función emparejarVectoresTF(
vectorTFQ: Vector TF de la consulta
vectorTFD: Vector TF del documento )
inicio
acumNumerador= 0
acumrDenominador= 0
#Se halla similitud de coseno
para i= vectorTFQ.length-1 hasta 0 con decremento 1 hacer
    datoQ= vectorTFQ[i];
    datoD= vectorTFD [i];
    acumrNumerador= acumNumerador+ (datoQ*datoD)

```

```

    acumDenominador= acumDenominador+(raizCuad(datoQ2)*raizCuad
(datoD2))
Fin para
retornar acumuladorNumerador/ acumuladorDenominador
Fin Función emparejarVectoresTF

```

La similitud de coseno permite medir el ángulo entre dos vectores en el espacio vectorial. Para la Ecuación 1, p es el vector del documento, q el del query, pi es el peso del término i dentro del documento y qi es el peso del término i del query (Jaramillo & Londoño, 2014).

Ecuación 1
Similitud de Coseno

$$\text{SimilitudCos}(\text{vectorDoc}, \text{vectorQ}) = \frac{\sum_{i=1}^n (p_i * q_i)}{\sqrt{\sum_{i=1}^n p_i^2} * \sqrt{\sum_{i=1}^n q_i^2}}$$

La función emparejarGeneral, ver Figura 4, se encarga de hallar la similitud de coseno entre el query y todos los documentos del corpus que contienen al menos un término ontológico de los presentes en el query. Para identificar estos documentos se hace uso del método descrito en la Figura 5, llamado hallarDocumentosCorpusRelevantesAcordeOntologia, que se apoya en una tabla hash correspondiente a un índice invertido en donde la clave es el término ontológico (y/o combinación de ellos) y la clave son los documentos relacionados.

Figura 4
Función emparejarGeneral

```

Funcion emparejarGeneral (
vectorTFQ: Vector TF de la consulta
vectorDocumentosCorpus: Listado documentos corpus
seleccionados)
para i= vectorDocumentosCorpus.length-1 hasta 0 con decremento
1 hacer
    documento= vectorDocumentosCorpus [i];
    similitud = emparejarVectoresTF(documento, vectorTFQ )
    listadoSimilitudes[i]= similitud
Fin para
retornar efectuarRanking (listadoSimilitudes)
Fin Función emparejarGeneral

```

Figura 5
Función hallarDocumentosCorpusRelevantesAcordeOntologia

```

Funcion hallarDocumentosCorpusRelevantesAcordeOntologia
(tablaHash: Indice Inverso documentos del
corpus
vectorTerminosConsultas: Vector con ids de términos ontológicos
que tiene la consulta,
vectorTFQ: Vector TF de la consulta)

documentosRelevantes=[ ]
i=0
para i= vectorTerminosConsultas.length-1 hasta 0 con decremento
1 hacer
    termino= vectorTerminosConsultas [i];
    #se buscar el termino ontologico para ver si devuelve un valor
    valor=tablaHash. obtener(termino)
    Si valor != null entonces

agregarDocumentosRelevantesAVectorDocumentosCoinciden
(valor)
    FinSi
Fin para
retornar efectuarRanking (vectorTFQ, documentosCoinciden)

```

El orden en que se muestran los documentos se obtiene mediante un ranking, en el que se ordenan de mayor a menor las medidas de similitud calculadas. Para este ranking se da prioridad a los documentos que tienen mayor cantidad de términos ontológicos. Esto quiere decir que si un documento tiene más términos ontológicos este se listará en primer lugar.

4. Resultados

El prototipo se desarrolla en Java. Se usan las librerías Jena y Lucene Analyzer. Jena (The Apache Software Foundation, 2019 a) permite el trabajo con ontologías (The Eclipse Foundation, 2019) y permite la creación de consultas sobre la ontología haciendo uso de SPARQL. Con Lucene analyzer (The Apache Software Foundation, 2019 b) se hallan las raíces de los términos mediante el Algoritmo de Porter. El corpus usado corresponde a The Colorado Richly Annotated Full Text Corpus (CRAFT) CRAFT V2.0 (Verspoor, et al., 2018), que consta de 67 artículos anotados semánticamente con la ontología Cell Ontology (CL).

Para la generación del índice invertido, que se almacena en una tabla Hash, se toma como entrada un documento anotado semánticamente. Este archivo es procesado para poder obtener los términos ontológicos. Ello se hace mediante el uso de expresiones regulares y de operaciones para gestión de cadenas. Las expresiones regulares se basan en el uso de patrones para hallar una combinación específica de caracteres dentro de un patrón utilizado para encontrar una determinada combinación de caracteres dentro de un String. Como resultado se obtiene un índice invertido que mapea cada término ontológico con una serie de documentos relacionados semánticamente. Esto permite hacer más eficientes las búsquedas. Se da prioridad a la búsqueda de términos ontológicos, pero en caso que el query no contenga ninguno de ellos, se halla similitud de coseno de forma clásica, tomando en consideración los términos hallados y sus correspondientes frecuencias.

Previo a la evaluación del sistema, se efectúa un recorrido a la ontología CRAFT (Verspoor, et al., 2018). Esto permite generar un listado (tabla *hash*) con todos los términos ontológicos. Por cada término se almacena el identificador y la etiqueta ontológica. A partir de este listado se elige aleatoriamente para la evaluación un total de 20 términos ontológicos. Las clases de interés que se tomaron en consideración para la evaluación se presentan en la Tabla 1. El corpus sobre el cual se realizan las pruebas está etiquetado con el estándar de oro.

Tabla 1
Clases de interés para la evaluación

#	Identificador	Etiqueta exacta dentro de la ontología	#	Identificador	Etiqueta exacta dentro de la ontología
1	CL_0000344	non-visual cell	11	CL_0000589	Inner hair cell
2	CL_0000345	dental papilla cell	12	CL_0000216	Sertoli cell
3	CL_0000457	biogenic amine secreting cell	13	CL_0000228	multinucleate cell
4	CL_0000346	hair papilla cell	14	CL_0000445	apoptosis fated cell
5	CL_0000066	epithelial cell	15	CL_0000287	eye photoreceptor cell
6	CL_0000187	muscle cell	16	CL_0000347	scleral cell
7	CL_0000561	amacrine neuron	17	CL_0000125	glial cell
8	CL_0000636	Muller cell	18	CL_0000178	Leydig cell
9	CL_0000125	glial cell			

Se generaron consultas para cada identificador. Se consideraron variantes tales como singular – plural, raíces, cambios en el orden de los términos y uso de relaciones de sinonimia. Para evaluar la calidad de los resultados se utilizó FileSeek (Binary Fortress Software, 2019). Esta herramienta permitió hallar los archivos que tenía el identificador del término ontológico buscado. Este listado se usó para verificar si los documentos mostrados en el ranking tenían la clase de oro buscada, y por lo tanto se consideran como verdaderos positivos. Las medidas de performance calculadas fueron accuracy, recall, precisión, medida F y área ROC. Los resultados obtenidos se presentan en la Tabla 2. Los resultados muestran que el sistema tiene buen performance, con lo que se concluye que el clasificador tiene un buen valor diagnóstico.

Tabla 2
Performance del sistema de búsqueda

Métrica de performance	Descripción	Fórmula	Valor obtenido para el clasificador
Exactitud (Accuracy)	Porcentaje de la data clasificada correctamente	$(TP+TN)/Total$	0.842
Recall (Tasa de verdaderos positivos)	La clase es positiva, ¿qué porcentaje logra clasificar?	$TPR = TP/(TP+FN)$	0.83
Precisión	Cuando predice positivos, ¿qué porcentaje clasifica correctamente?	$TP/(TP+FP)$	0.842
F-Measure	Media armónica de precisión y Recall	$(2*precision*recall)/(precision+recall)$	0.8531
ROC			0.9237

5. Conclusiones

La técnica de búsqueda presentada en este artículo se apoya en Web semántica para proporcionar al usuario documentos relevantes acorde a su necesidad de búsqueda. El uso de ontologías permite la expansión del query y valerse de relaciones tales como sinonimia, logrando con ello dar mayor importancia al contexto. La aproximación presentada usa la representación TF de los documentos y se apoya en índices invertidos para mostrar los resultados al usuario.

Este sistema ofrece ventajas tales como: más similitud con el query, más velocidad a la hora de entregar documentos debido al proceso de indexación y precisión más alta gracias a la expansión del query.

En la actualidad, sobre un corpus estático, la creación de los índices invertidos se crea una sola vez. Se plantea como trabajo futuro actualizar incrementalmente el índice a medida que se agreguen documentos al corpus. Los resultados experimentales muestran que el sistema es eficaz para recuperar información.

Referencias bibliográficas

- AGRAWAL, R., & SRIKANT. (1995). Mining sequential patterns. *Proceeding ICDE '95 Proceedings of the Eleventh International Conference on Data Engineering*, 3-14.
- ANDERSON, J. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*. 22(3), 261-295.
- BINARY FORTRESS SOFTWARE. (2019). *Fileseek. Fast and Free File Search*. Recuperado el Octubre de 2019, de <http://www.fileseek.ca/>

- BORST, W. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Ph.D. Dissertation, University of Twente.
- CEGLAR, A., & RODDICK. (2006). Association mining. ACM Computing Surveys (CSUR) Surveys Homepage archive. *38(2)*, artículo 5.
- FELLBAUM, C. (2005). WordNet and wordnets. En Encyclopedia of Language and Linguistics. 665-670.
- JARAMILLO, S., & LONDOÑO, J. (2014). Búsqueda de documentos basada en el uso de índices ontológicos creados con MapReduce . *Ciencia e Ingeniería Neogranadina*, *24(2)*.
- JIANG, X., & TAN, A. (2006). OntoSearch: a full-text search engine for the semantic web. (págs. 1325-1330). AAAI Press. Obtenido de <http://dl.acm.org/citation.cfm?id=1597348.1597399>
- PEREZ, A. (2016). Semantic Web and Semantic Technologies to enhance Innovation and Technology Watch processes. En *Mondragon Goi Eskola Politeknikoa Electronics and Computing Department*.
- PERINAN-PASCUAL, C., & ARCAS-TUNEZ, F. (2014). La ingeniería del conocimiento en el dominio legal: La construcción de una Ontología Satélite en FunGramKB. *Revista signos*, *47*, 113-139. Retrieved from https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-09342014000100006&nrm=iso
- SHAMSFARD, M., NEMATZADEH, A., & SARAH, M. (2006). ORankAn Ontology Based System for Ranking Documents. *International Journal of Electrical and Computer Engineering*, *1:3*. Obtenido de ORank: An Ontology Based System for Ranking Documents.
- THE APACHE SOFTWARE FOUNDATION. (2019). *Apache Jena*. Recuperado el 6 de octubre de 2019, de <http://jena.apache.org/>
- THE APACHE SOFTWARE FOUNDATION. (2019). *Apache Lucene Core*. Recuperado el 6 de octubre de 2019, de <http://lucene.apache.org/core/>
- THE ECLIPSE FOUNDATION. (2019). *Eclipse Project*. Recuperado el octubre de 2019, de <http://www.eclipse.org/>
- VERSPoor, K., COHEN, K., LANFRANCHI, A., WARNER, C., JOHNSON, H., ROEDER, C. (2018). Craft 2.0. Obtenido de BioNLP-Corpora: <http://sourceforge.net/projects/bionlp-corpora/files/CRAFT/v0.9/>
- WANG, L. & TSAI. (2008). A practical ontology query expansion algorithm for semantic-aware learning objects retrieval. *Computers Education*, *50(4)*, 1240-1257.
- YUNIEL, P., & ALIANIS, P. (2012). OntoCatMedia: ontología para la búsqueda y clasificación automática de medias audiovisuales. *Ciencias de la Información*, *43*, 49-54. Retrieved from <https://www.redalyc.org/articulo.oa?id=181424691006>
- ZAKI, M. (2000). Scalable algorithms for association mining. *IEEE Trans.Knowl.Data Eng*, *12(3)*, 372-390.
- ZOU, G. Z. (2008). An Ontology-Based Methodology for Semantic Expansion Search. FSKD '08: Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery. *5*, 453-457.

-
1. Doctora en Ingeniería. Programa de Ingeniería de Sistemas y Computación, Universidad del Quindío, UQ, Armenia, (Colombia), sjaramillo@uniquindio.edu.co
 2. Magister en Ingeniería. Programa de Tecnología en Topografía. Universidad del Quindío, UQ, Armenia, (Colombia)
 3. Magister en Ingeniería. Programa de Ingeniería de Sistemas y Computación. Universidad del Quindío, UQ, Armenia, (Colombia)
-

Revista ESPACIOS. ISSN 0798 1015
Vol. 41 (Nº 15) Año 2020

[\[Índice\]](#)

[En caso de encontrar algún error en este website favor enviar email a [webmaster](#)]

revistaESPACIOS.com



This work is under a Creative Commons Attribution-
NonCommercial 4.0 International License