SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

Tecnológico Nacional de México
Instituto Tecnológico de Ciudad Madero

División de Estudios de Posgrado e Investigación

Implementación de un prototipo de un sistema de recuperación de información que utilice ontologías para la expansión de consultas

TESIS

Para obtener el grado de:
Maestro en Ciencias de la Computación

PRESENTA:
ISC Lucía Janeth Hernández González

DIRECTOR:
Dr. Juan Javier González Barbosa

Cd. Madero, Tamaulipas, México

Octubre 2016

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Ciudad Madero

Cd. Madero, Tamps; a 29 de Septiembre de 2016.

OFICIO No.: U5.175/16
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

ING. LUCÍA JANETH HERNÁNDEZ GONZÁLEZ
NO. DE CONTROL G09070934
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias de la Computación, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	DRA.	MARÍA LUCILA MORALES RODRÍGUEZ
SECRETARIO :	DR.	JOSÉ ANTONIO MARTÍNEZ FLORES
VOCAL :	DR.	JUAN JAVIER GONZÁLEZ BARBOSA
SUPLENTE	DRA.	CLAUDIA GUADALUPE GÓMEZ SANTILLÁN
DIRECTOR DE TESIS:	DR.	JUAN JAVIER GONZÁLEZ BARBOSA

Se acordó autorizar la impresión de su tesis titulada:

"IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA DE RECUPERACIÓN DE INFORMACIÓN QUE UTILICE ONTOLOGÍAS PARA LA EXPANSIÓN DE CONSULTAS"

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta.

Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A T E N T A M E N T E
"POR MI PATRIA Y POR MI BIEN"®

DRA. ADRIANA ISABEL REYES DE LA TORRE
JEFA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN

S.E.P.
DIVISION DE ESTUDIOS
DE POSGRADO E
INVESTIGACION
I T C M

c.c.p.- Archivo
Minuta

AIRT'NLCO ' amhl

Ave. 1° de Mayo y Sor Juana I. de la Cruz Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Tel. (833) 357 48 20. e-mail: itcm@itcm.edu.mx
www.itcm.edu.mx

Declaración de originalidad

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones.

Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

Octubre 2016, Cd. Madero, Tamaulipas

ISC Lucía Janeth Hernández González

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Tecnológico Nacional de México sede Instituto Tecnológico de Ciudad Madero por su patrocinio para la realización y culminación de este proyecto de tesis titulado Ïmplementación de un prototipo de un sistema de recuperación de información que utilice ontologías para la expansión de consultas".

De igual manera agradezco al Doctor Juan Javier González Barbosa por la confianza brindada a lo largo de estos últimos dos años, demostrándome siempre una excelente actitud y apoyo incondicional. Así como el otorgarme el consejo y la asesoría necesaria para alcanzar las metas establecidas en este trabajo de tesis.

De igual forma agradezco a la Dra. Claudia Guadalupe Gómez Santillán, a la Dra. María Lucila Morales Rodríguez y al Dr. José Antonio Martínez Flores por sus más finas atenciones a lo largo de este período.

También agradezco a mi familia por el integro apoyo concedido a lo largo de mi carrera profesional, el cual propicio en la culminación de esta nueva etapa académica. Enseñándome que la familia es la base del éxito del mañana.

Así mismo a mis amigos y compañeros que me atribuyeron su apoyo absoluto originando una adecuada estabilidad laboral y personal, lo cual auxilio a mi desarrollo académico.

Resumen

Los Sistemas de Recuperación de Información (IRS) son un estudio importante desde 1945 [Singhal, 2001]. La esencia de estos sistemas está en el empleo de los Modelos de Recuperación de Información (IRM), los cuales se encargan de comparar los términos semejantes de consulta con una colección de documentos tomando en cuenta la concurrencia de términos, para recuperar un conjunto de documentos de acuerdo a ciertos criterios de clasificación. Existen métricas que permiten clasificar un conjunto de documentos de acuerdo al grado de similitud, tal como el coseno de similitud y el *soft cosine measure* [Sidorov et al., 2014].

En este trabajo se compararon tres diferentes métodos de expansión:(a) Familia léxica, (b) sinónimos y (c) lexema, utilizando el dominio en Ciencias de la Computación. El IRM seleccionado para realizar la búsqueda y recuperación de documentos en este trabajo fue el Modelo de Espacio Vectorial (VSM), el cual empleo una muestra de la colección `CACM`. Esta muestra está compuesta por 5 consultas y 130 documentos.

Y para determinar cual de las dos métricas de similitud mencionadas en el primer párrafo sería aplicada en el proceso de clasificación se realizo una experimentación para evaluar los índices de *Recall* y *Precision*. Ambas métricas obtuvieron el 100 % de *Recall* lo cual indica que el sistema recupero todos los documentos relevantes en cada consulta. Sin embargo *Soft Cosine Measure* obtuvo 0.7 % más precisión que la métrica del coseno, por lo tanto se utilizo este último.

Para hacer las expansiones de las consultas se agregaron por cada término de la consulta un máximo de 5 términos que comparte el mismo lexema para la familia léxica; un máximo de 5 sinónimos para la expansión usando sinónimos y para la expansión del lexema se incorporó por cada término de la consulta, el lexema y en caso de que el término sea un lexema no se agrega ningún término.

Los resultados obtenidos en la fase experimental bajo la configuración de 5 consultas y 130 documentos fueron que el *Recall* fue de un 100 % para todas las consultas. Utilizando los resultados de precisión se realizo una prueba de Wilcoxon que muestra diferencia estadísticamente significativa del 0.0411 entre la consulta original y el método de expansión con sinónimos.

Summary

The information retrieval models (IRM) have been of important study since 1945 [Singhal, 2001]. The essence of these is in the use of Information Retrieval Models (IRS), which are in charge of compare a query and a set collection of documents, taking into consideration the concurrent terms to retrieval a set of document according to the clasification criterion. There are metrics to classify a set of documents according to the grade of similarity, such as similarity cosine and soft cosine measure [Sidorov et al., 2014].

In this paper three different expansion methods were compared: (a) Lexical Family, (b) synonyms and (c) lexeme using the domain in Computer Science. The IRM selected for search and retrieval of documents in this work was the vector space model (VSM), which use a sample collection `CACM`. This sample consists of 5 documents and 130 queries.

And to determine which of the two similarity metrics mentioned in the first paragraph would be applied in the classification process an experiment was conducted to evaluate the indices *Recall* and *Precision*. Both metrics obtained 100 % of *Recall* which indicates that the system recovered all relevant documents for each query. However *Soft* obtained Cosine Measure 0.7 % more accurately than the cosine metric, therefore the latter was used.

To make consultations expansions were added for each query term a maximum of 5 terms that share the same lexeme for lexical family; a maximum of 5 synonyms for expansion using synonyms and expansion of the stem joined by each query term, the stem and if the term is a lexeme no term is added.

The results obtained in the experimental phase under configuration of 5 consultations and 130 documents were the *Recall* was 100 % for all queries .By using precision results Wilcoxon test showing statistically significant difference of 0.0411 between the query original and the method of expansion with synonyms.

Contenido general

Lista de Figuras

Lista de Tablas

Lista de Algoritmos

Capítulo 1

Introducción

Internet es actualmente una de las principales fuentes de información para la mayoría de las personas. A pesar de que el gran volumen de información existente en Internet refleja una infinidad de ventajas para los usuarios, éste ha provocado que encontrar información o recursos deseados sea más difícil y ambiguo.

Los motores de búsqueda web son utilizados para buscar información en Internet, pero debido a la gran cantidad de información disponible, en algunas ocasiones muestran al usuario resultados desfavorables, es decir, no muestran la debida información que el usuario necesita. Por tal motivo, la necesidad de técnicas, modelos o sistemas que permitan obtener información relevante y exacta es ya un hecho que acongoja actualmente a los internautas y a los investigadores que buscan disminuir esta problemática.

Aunque los motores de búsqueda web han evolucionado con el objetivo de mejorar la recuperación de información, esto no ha sido suficiente debido al constante incremento de información. Estos motores utilizan modelos que permiten extraer y mostrar información con cierto grado de prioridad dada una consulta, conocidos como Modelos de Recuperación de Información (IRM). La lógica de estos modelos se basa en la clasificación de información en base a la concurrencia de términos.

Existen modelos que permiten mejorar la búsqueda y clasificación de información, tal es el Modelo de Espacio Vectorial (VSM), aunque no fue el primero en su rama si ha sido el más adoptado por los investigadores por la eficiencia y sencillez ante la mayoría de los modelos. Este modelo hace uso de matrices y vectores, técnicas de normalización de términos, entre otros, para obtener una mejor aproximación a la información que el usuario necesita.

Los IRS se basan en la concurrencia de términos, lo cual limita la precisión en los procesos de búsqueda y la clasificación de información. Desde hace varios años se implementan técnicas o herramientas para disminuir esta carencia, como lo son las ontologías, que en su definición informal son una herramienta que permite la representación de conocimiento.

Las ontologías permiten contener en su estructura información conceptual sobre un cierto dominio, permitiendo tener una pequeña muestra del lenguaje natural de un tema en específico. Esta característica puede llegar a disminuir la ambigüedad en la

recuperación de información, ya que proporciona una perspectiva más amplia de lo que el usuario quiere buscar. Se puede decir que mientras más enriquecida y precisa esté una consulta, podría mejorar la extracción de información.

Este trabajo contempla mejorar la precisión de las consultas mediante tres diferentes expansiones catalogadas como: familia léxica, sinónimo y lexema. Estas agregan términos a la consulta bajo diferentes criterios. La familia léxica de un término de la consulta es obtenida mediante una ontología. Las expansiones con sinónimos y lexema son obtenidas mediante un corpus de la herramienta *Freeling*.

Dicho lo anterior, en esta investigación se buscó expandir una serie de consultas con la ayuda de una ontología con dominio en Ciencias de la Computación. Permitiendo ampliar y mejorar la cobertura de búsqueda al IRS.

La distribución de la información en este trabajo de tesis se compone como sigue: el primer capítulo contiene el planteamiento del problema, la justificación, el objetivo general y específicos, así como los alcances y limitaciones. El capítulo dos contiene información teórica que permite despejar las dudas sobre los tecnicismo de este trabajo. El capítulo tres se describen y comparan los trabajos relacionados con esta tesis. El capítulo cuatro describe la implementación del VSM y la ontología. El capítulo cinco muestra los resultados de las diferentes expansiones que se realizaron de las consultas. Finalmente, el capítulo seis muestra las conclusiones y el trabajo futuro de este trabajo.

1.1. Planteamiento del problema

La problemática de realizar la búsqueda con los Sistemas de Recuperación de Información (IRS) ha crecido debido al incremento colosal de la información. Estos sistemas, que a menudo se relacionan con los buscadores web, permiten al usuario visualizar y/u obtener la información de acuerdo a una consulta. Este procedimiento a menudo se realiza principalmente con un IRM que, mediante la extracción de palabras clave en la consulta aplicando técnicas de procesamiento de lenguaje natural, indexan una serie de documentos que tienden a ser aquellos con los porcentajes más altos de concurrencia. Este mecanismo en sistemas con grandes volúmenes de información tiende a ser limitado aunque implemente uno o varios IRM's.

En general, al procesar una consulta en un IRM tiende a no mostrar la información necesaria que el usuario busca. Normalmente se presentan dos problemáticas de clasificación: falsos positivos y falsos negativos. Los falsos positivos son los que en ocasiones recuperan demasiada información que no es significativa para el usuario. Los falsos negativos son los que no incorporan en el resultado documentos que deberían ser parte de la información mostrada al usuario.

Con el fin de minimizar estas problemáticas, se aplicaron tres diferentes métodos de expansión de consultas con dominio en ciencias de la computación, con el fin de generar consultas con una semántica más enriquecida. Estas expansiones permitirán incorporar a las consultas términos que contengan similitudes semánticas con los términos originales. Estas contendrán la familia léxica, los sinónimos y lexemas de cada término que se manejen con mayor frecuencia e importancia dentro de la naturaleza de las colecciones o corpus de documentos, permitiendo así la expansión de la consulta.

1.2. Justificación

La evolución de las tecnologías y las ciencias la información en Internet ha incrementado exponencialmente, ocasionando que sea más difícil encontrar la información de acuerdo a nuestros intereses. Con el fin mejorar la obtención de información en Internet surgieron los IRM, los cuales proporcionan de manera estadística o probabilística la relevancia de los documentos dentro de una colección dada una consulta.

Sin embargo, estos modelos tienden a ser limitados debido a que la esencia de estos está basada en el léxico de las palabras que contengan las consultas y los documentos. Esto quiere decir que los documentos clasificados como "relevantes" serán solo aquellos que contengan exactamente la mayor frecuencia de las palabras contenidas en la consulta. En otras palabras, estos modelos no contienen un procesamiento conceptual, por lo que están limitados a los términos de la consulta. Esto puede ocasionar que documentos irrelevantes que compartan términos con la consulta sean recuperados y clasificados aún mejor que los relevantes.

En este contexto, las expansiones de consultas han incrementado su uso en la recuperación de información debido a que permiten representar la conceptualización de uno o varios dominios (vocabulario de los temas). Se espera que la incorporación de una ontología en este proceso de clasificación mejore la precisión de los resultados.

El beneficio que se obtiene de este trabajo de investigación es implementar un IRS que permita expandir las consultas de un usuario utilizando diferentes métodos de expansión. Esto permitirá mejorar la limitante que tienen los IRS y con esto mostrar en cierta medida una mejor calidad y precisión en el mecanismo de clasificación.

1.3. Objetivos del Proyecto

1.3.1. Objetivo general

Implementar la expansión de consultas utilizando una ontología con dominio en Ciencias de la Computación en un prototipo de sistema de recuperación de información utilizando colecciones de documentos y consultas estándar.

1.3.2. Objetivos específicos

Los objetivos específicos alcanzados en este trabajo de tesis para poder cumplir el objetivo general se listan a continuación:

- Buscar y definir el dominio a conceptualizar en la ontología, así como el IRM.
- Buscar y establecer las colecciones de documentos y consultas estándar a utilizar
- Crear una ontología con el dominio elegido.
- Implementar un IRS.

- Generar un nivel de 5 expansiones por consulta.
- Realizar experimentos.
- Reportar los resultados.

1.4. Alcances y limitaciones

Los alcances del presente trabajo se listan a continuación:

- Utilizar técnicas de procesamiento de lenguaje natural para análisis léxico de las consultas y para la colección de documentos.
- El dominio de la ontología será en Ciencias de la Computación para realizar la expansión de las consultas.
- Generar un máximo de 5 expansiones por término a partir del original.
- Emplear el modelo de recuperación de información de Espacio Vectorial.
- Clasificar los documentos recuperados mediante la métrica de similitud *Soft Cosine Measure* [Sidorov et al., 2014].
- Construir el IRS y generar la expansión de consultas empleando el lenguaje de programación Python 3.4.
- Emplear el IDE Protégé para la modificación, estructuración y visualización de la ontología.
- Construir la ontología mediante el lenguaje de programación Java.

Las limitantes del mismo se muestra a continuación:

- El idioma a manejar sobre las consultas y la colección de documentos será estrictamente en inglés.
- Sólo se manejará una colección de documentos estándar, con una cantidad no mayor a 1000 documentos.
- Las consultas a manejar deberán pertenecer a la colección de documentos que se elija.

Capítulo 2

Marco teórico

2.1. Internet

El Internet es el más claro ejemplo de un medio de acceso, recuperación y visualización de información, este cuenta con la mayor demanda y disponibilidad que hay en la actualidad. La consolidación de la web como plataforma de diseño de los sistemas de información en Internet propició la creación de los IRS más voluminosos y avanzados que hasta ahora se han desarrollado [Martínez Méndez, 2004].

Dentro de la magnitud y dimensión del Internet, la búsqueda y recuperación de documentos de diferentes dominios es uno de los servicios con mayor demanda conocido como la World Wide Web (WWW o web), que de acuerdo a [Miniwatts Marketing Group., 2015] es un conjunto de protocolos que permiten la consulta remota de archivos de hipertexto.

2.1.1. Tamaño o dimensión de WWW

Comúnmente se suele confundir que los términos de Internet y WWW son lo mismo, pero como se mencionó en la sección anterior, la WWW es un servicio que proporciona el Internet. Tan solo este servicio suministra un enorme volumen de información.

El impacto suscitado en los últimos años de WWW en la sociedad ha generado un incremento masivo de este, tanto que hasta la fecha no se puede dar un valor exacto sobre el tamaño o dimensión del mismo. Sin embargo, un estudio realizado por [Maurice de Kunder, 2015] muestra una estimación de la dimensión de la WWW del periodo Diciembre 2014 a Febrero 2015, la cual muestra que es de al menos 4.63 miles de millones de páginas (Figura 2.1). Tal estimación se basa sobre el número de páginas indexadas por *Google*, *Bing* y *Yahoo Search*.

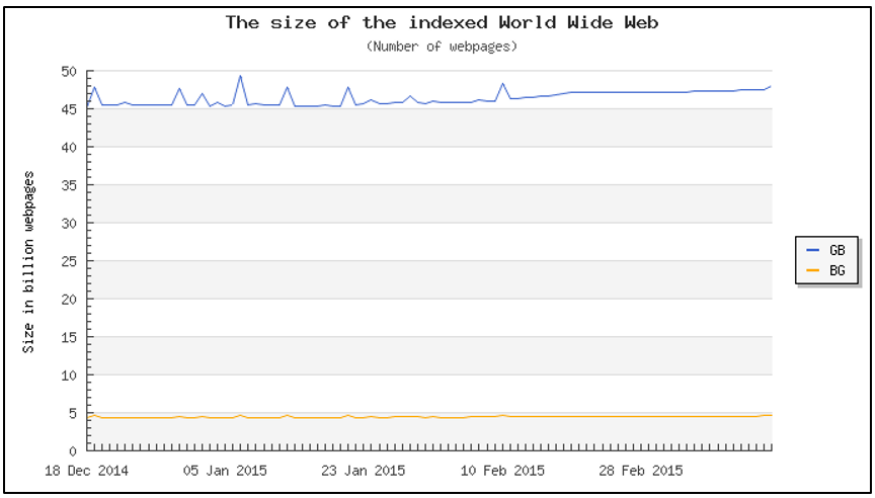

Figura 2.1: Tamaño estadístico de la WWW

Ademas Kunder explica que el tamaño mínimo estimado del índice WWW en la Figura 2.1 se basa en las estimaciones del número de páginas indexadas por *Google*, *Bing* y *Yahoo Search*. A partir de la suma de estas estimaciones, una superposición estimada entre estos motores se resta. La superposición es una sobreestimación; por lo tanto el tamaño total estimado del índice WWW es una subestimación.

2.1.2. Número de usuarios

Así como la información contenida dentro del conjunto de protocolos WWW incrementa de manera exponencial, así mismo lo hace el número de usuarios que día a día utilizan el Internet para consultar este gigantesco servicio. De acuerdo a estadísticas realizadas por [Miniwatts Marketing Group., 2015] sobre la cifra estadística del número de usuarios que navegan por Internet, muestran que al 30 de Noviembre del 2015 había un total de **3,366,261,156** usuarios.

Tal estudio se puede observar en la Tabla 2.1, la cual en la primera columna tiene las regiones del mundo donde se realizaron los estudios. En la segunda columna muestra la cantidad de personas que integran la población de esa región. En la tercera columna muestra que porcentaje representa dicha población a nivel mundial. En la cuarta columna se observa que cantidad de usuarios tiene esa región hasta el 30 de noviembre del 2015, en la quinta columna indica el porcentaje de la población que forman parte de los usuarios de Internet estimados para esa región. En la sexta columna muestra el incremento de usuarios desde el 2000 hasta el 2015. Finalmente en la última columna muestra que porcentaje abarca la cantidad de usuarios de esa región en comparación del tamaño total de usuarios en Internet.

Tabla 2.1: Porcentaje de la población que utiliza Internet [Miniwatts Marketing Group., 2015].

INTERNET MUNDIAL USADO Y ESTADÍSTICAS DE LA POBLACIÓN **NOVIEMBRE 30, 2015**						
Regiones del Mundo	**Población (2015 Est.)**	**% de Población mundial**	**Usuarios en Internet 30-nov-2015**	**Penetración (% Población)**	**Crecimiento 2000-2015**	**% de usuarios**
África	1,158,355,663	16.0 %	330,965,359	28.6 %	7,231.3 %	9.8 %
Asia	4,032,466,882	55.5 %	1,622,084,293	40.2 %	1,319.1 %	48.2 %
Europa	821,555,904	11.3 %	604,147,280	73.5 %	474.9 %	18.0 %
Medio Oriente	236,137,235	3.3 %	123,172,132	52.2 %	3,649.8 %	3.7 %
América del Norte	357,178,284	4.9 %	313,867,363	87.9 %	190.4 %	9.3 %
America Latina / Caribe	617,049,712	8.5 %	344,824,199	55.9 %	1,808.4 %	10.2 %
Oceanía / Australia	37,158,563	0.5 %	27,200,530	73.2 %	256.9 %	0.8 %
TOTAL MUNDIAL	7,259,902,243	100.0 %	**3,366,261,156**	46.4 %	832.5 %	100.0 %

Como se puede apreciar en la Tabla 2.1 la cantidad de usuarios en Internet es del 46.4 % del total de la población mundial. También se puede predecir que este número va a incrementar con el paso de los años, de acuerdo al crecimiento notable del 2000 al 2015.

2.1.3. Número de usuarios en Internet por lenguaje

Estimar el porcentaje de los idiomas que encabezan los primeros lugares de concurrencia en los usuarios permite determinar que lenguaje es el más apropiado abordar en el proceso de IR. Nuevamente, el estudio realizado por [Miniwatts Marketing Group., 2015] se conoce un estimado del top diez de los lenguajes más empleados en Internet.

En la Tabla 2.2, observamos cuales son los diez idiomas más usados en Internet, donde la columna dos indica el número de usuarios que navegan en Internet con ese idioma. En la tercera columna muestra el cociente entre la suma de los usuarios de Internet que hablan una lengua y la estimación total de la población que habla ese idioma específico. La cuarta columna indica el porcentaje de incremento de usuarios en Internet en el intervalo del 2000 - 2015 La quinta columna señala el porcentaje que

representa el número de usuarios con ese idioma a nivel mundial. Finalmente la última columna indica la cantidad de personas que hablan en ese idioma.

Tabla 2.2: Promedio de los 10 idiomas más utilizados en Internet [Miniwatts Marketing Group., 2015].

Top 10 de los Lenguajes utilizados en la Web. 30 de Noviembre, 2015 (Número de usuarios en Internet por Lenguaje)					
TOP 10 Lenguajes en Internet	**Usuarios Internet por Lenguaje**	**Penetración Internet (% Poblacion)**	**Crecimiento usuarios en Internet (2000 - 2015)**	**% Total mundial de usuarios en Internet (Participación)**	**Población mundial para este lenguaje (2015 Estimación)**
Ingles	**872,950,266**	62.4 %	520.2 %	**25.9 %**	1,398,283,969
Chino	704,484,396	50.4 %	2,080.9 %	20.9 %	1,398,335,970
Español	**256,787,878**	58.2 %	1,312.4 %	**7.6 %**	441,052,395
Árabe	168,176,008	44.8 %	6,592.5 %	5.0 %	375,241,253
Portugués	131,903,391	50.1 %	1,641.1 %	3.9 %	263,260,385
Japones	114,963,827	90.6 %	144.2 %	3.4 %	126,919,659
Ruso	103,147,691	70.5 %	3,227.3 %	3.1 %	146,267,288
Malacia	98,915,747	34.5 %	1,626.3 %	2.9 %	286,937,168
Francés	97,729,532	25.4 %	714.9 %	2.9 %	385,389,434
Alemán	83,738,911	87.8 %	204.3 %	2.5 %	95,324,471
TOP 10 Lenguajes	2,632,248,147	53.5 %	787.0 %	78.2 %	4,917,011,992
Resto de los lenguajes	734,013,009	31.3 %	1,042.9 %	21.8 %	2,342,890,251
TOTAL MUNDIAL	3,366,261,156	46.4 %	832.5 %	100.0 %	7,259,902,243

Como se puede observar en la Tabla 2.2 el idioma en Inglés representa la mayor tendencia del habla en Internet con un 25.9 % dando una cantidad de **872,950,266** usuarios, dejando por mucho al idioma español con **256,787,878** usuarios, que representa el 7.6 % del porcentaje total de usuarios en Internet.

2.2. Procesamiento de lenguaje natural (NLP)

NLP es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. Por esta razón es importante para una amplia gama de personas tener un conocimiento práctico de NLP.

Dentro de la industria, esto incluye a las personas en la interacción computadora – humano, análisis de información de negocios y desarrollo de software web. Dentro de la Academia, este incluye a personas en áreas de computación humana y corpus lingüísticos a través de la ciencia computacional e inteligencia artificial [Bird et al., 2009]. Algunos de los mecanismos más utilizados en NLP son los descrito en las secciones 2.2.2, 2.2.3 y 2.2.4. Sin embargo es posible encontrar dichos mecanismos para su utilización en una herramienta conocida como NLTK.

2.2.1. NLTK (*Natural Language Toolkit*)

NLTK es una herramienta que permite trabajar con los datos del lenguaje humano. Proporciona interfaces fáciles de usar para más de 50 corpus y recursos léxicos como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para la clasificación, tokenización, derivaciones (*stemming*), etiquetado (*tagging*), análisis y de razonamiento semántico y contenedores para las bibliotecas de NLP de potencia industrial. Las bibliotecas más comunes de NLTK para NLP son:

- Tokenizacion.
- Etiquetado gramatical (*Part-of-speech tagger*):

Un *token* es el nombre técnico para una secuencia de caracteres tales como ***hair***, ***it*** o ***:)*** [Bird et al., 2009]. Entonces, la tokenizacion envuelve la separación de una oración en una secuencia de caracteres, tal secuencia puede componerse de diferentes patrones dado que un *token* puede suscitarse como sigue:

- *word*
- *word,*
- *:)*

El indicador default para separar una oración en *tokens* es el espacio en blanco, en caso de no tenerlo la oración será separada carácter por carácter. NLTK puede ser aplicado si el texto a procesar está en el idioma Ingles.

2.2.2. Palabras vacias (*stopwords*)

Para poder procesar e interpretar un conjunto de palabras de manera computacional es ideal que las palabras que compongan dicho conjunto ofrezcan información signifi-

cativa. Las palabras que carezcan de esto son conocidas como *stopwords*.

Un *stopword* puede ser identificada como una palabra que tiene la misma probabilidad de ocurrencia en aquellos documentos no relevantes para una consulta como en aquellos documentos relevantes para la consulta [Wilbur and Sirotkin, 1992]. En la Tabla 2.3 se pueden apreciar algunas de aquellas palabras catalogadas como *stopwords*.

Tabla 2.3: Conjunto representativo de palabras conocidas como *stopwords*, [Bird et al., 2009].

i - me - my - myself - we - our - ours - ourselves - you - your - yours - yourself - yourselves - he him - his - himself - she - her - hers - herself - it - its - itself - they - them - their - theirs themselves - what - which - who - whom - this - that - these - those - am - is - are - was - were be - been - being - have - has - had - having - do - does - did - doing - a - an - the - and - but if - or - because - as - until - while - of - at - by - for - with - about - against - between into - through - during - before - after - above - below - to - from - up - down - in - out - on off - over - under - again - further - then - once - here - there - when - where - why - how - all any - both - each - few - more - most - other - some - such - no - nor - not - only - own - same so - than - too - very - can - will - just - don - should - now

Por lo tanto, se puede concluir que las palabras que obtienen una probabilidad de ocurrencia menor a las *stopwords* pueden proporcionar la información necesaria para distinguir a los documentos relevantes de aquellos que no lo son. La identificación de *stopwords* en un texto permite apreciar y usar aquellas palabras que brindan de información sustancial.

2.2.3. Etiquetado gramatical (*POS tagging*)

El proceso de clasificación de las palabras en sus partes léxicas (*part-of-speech*) y etiquetarlos acorde con ello se conoce como etiquetado gramatical (*part-of-speech tagging*), etiquetado POS (*POS tagging*), o simplemente etiquetado (*tagging*). *Part-of-speech* también es conocido como clases de palabras o categorías léxicas. La colección de etiquetas utilizado para una tarea en particular se conoce como un conjunto de etiquetas [Bird et al., 2009].

POS tagging es una técnica de NLP la cual procesa un conjunto de palabras (o una palabra) y coloca una etiqueta en cada palabra con su respectivo etiquetado gramatical (tipo de palabra). En el ámbito computacional existen herramientas que integran este proceso de etiquetado, permitiendo emplearla de una manera más práctica y sencilla.

La herramienta NLTK contiene un corpus de etiquetado, este corpus contiene los diferentes tipos de clase que puede identificar y catalogar. En la Tabla 2.4 se puede apreciar algunos etiquetados gramaticales que contiene dicha herramienta, donde en la columna uno se indica la descripción del etiquetado y en la segunda columna se observa la abreviación de la descripción.

Tabla 2.4: Algunos etiquetados gramaticales que NLTK reconoce [Bird et al., 2009].

Descripción de la clase de palabra	***POS tagging***
noun, proper, singular	NNP
noun, common, singular or más	NN
noun common plural	NNS
numeral, cardial	CD
adjetivo or numeral, ordinal	JJ
verb past tense	VBD
adverb	RB
verb, present participle or gerund	VBG
verb, present tense, 3rd person singular	VBZ
verb, present tense, not 3rd person singular	VBP
verb, past participle	VBN
preposition or conjunction, subordinating	IN
noun, proper, plural	NNPS
verb, base form	VB
determiner	DT
adjetive, comparative	JJR
pronoun personal	PRP
modal auxiliary	MD
pronoun, possessive	PRP$
adjetive, superlative	JJS
WH-determiner(that, what, whatever, which, whichever)	WDT
conjunction, coordinating	CC

Como se puede apreciar en la Tabla 2.4 existen diferentes derivados de los componentes importantes de una oracion. Los cuatro etiquetados gramaticales que juegan un papel importante en el significado de una oración o de un texto completo son los **sustantivos**, **adjetivos**, **verbos** y **adverbios** [UNAM, 2015]. En la Figura 2.2 se observa un ejemplo de cómo aplica NLTK el *POS tagging*.

Para poder aplicar el *POS tagging* en una oración primeramente se tokeniza, la oración empleada en la Figura 2.2 que está señalada por el cuadro color rojo. Después se procede a identificar el tipo de palabra. El etiquetado de cada *token* se indica en los

```
>>> text = nltk.word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
('completely', 'RB'), ('different', 'JJ')]
```

Figura 2.2: *POS tagging* aplicado en una oracion

cuadros verdes. La descripción de cada etiqueta se puede encontrar en la Tabla 2.4.

2.2.4. Reducción de palabras a raíz (*stemming*)

Stemming es un método ampliamente utilizado en la normalización de textos diseñado para permitir el emparejamiento de términos relacionados morfológicamente, tales como *cluster* y *clustering*. La idea es que en un lenguaje como el Inglés, una palabra común contiene un *stem* (raíz) el cual hace referencia a alguna idea central o "significado", y que ciertos afijos (morfemas) han sido agregados para modificar el significado y/o para ajustar la palabra para ese rol sintáctico. El propósito del *stemming* es despojar el afijo y por lo tanto reducir la palabra a su esencia (*stem*) [Paice, 1994].

De acuerdo con la Real Academia Española, un afijo es un morfema que modifica el significado o propiedades gramaticales de la base (*stem*) a la que se adjunta. Un afijo está unido antes, después o dentro de un *stem*. Cuando un afijo es agregado al principio de una palabra *stem* se conoce como prefijo, cuando es agregado al final es conocido como sufijo [E. Loos et al., 2003]. Un ejemplo de los afijos más conocidos se muestran en la Tabla 2.5. En la columna uno se encuentra el tipo de afijo. En la columna dos esta una definición informal del afijo. Finalmente en la columna tres se encuentra un ejemplo del afijo.

Tabla 2.5: Ejemplos de algunos tipos de afijos.

Afijo	**Relación con el *stem***	**Ejemplo**
Prefijo	Colocado al principio	discontinuo
Sufijo	Colocado al final	Maquinaria
Infijo	Colocado en medio	frialdad
Circumfijo	Ocurre en dos partes, en ambos bordes exteriores	Anaranjar

Al conjunto de diferentes afijos que una palabra raíz puede adquirir se le conoce como familia léxica. De acuerdo a [Lobo, 2010] se entiende que familia léxica es el grupo de palabras que comparten el mismo lexema (raíz o *stem*) y por lo tanto un significado común.

Por ende en una familia léxica se reúnen todos los derivados, compuestos y parasintéticos de la palabra primitiva, que contengan el mismo lexema básico tanto patrimonial (si la palabra ha evolucionado fonológica-mente) como culto (si ha mantenido

su forma original latina). En la Tabla 2.6 se muestran algunas palabras en su estado raíz con su respectiva familia léxica. En la columna uno se encuentra la palabra raíz. En la segunda columna se encuentra la familia léxica de la palabra raíz.

Tabla 2.6: Ejemplo de algunas palabras raíz con sus familias léxicas.

Lexema, raiz o *stem*	**Familia léxica**
Arte	artístico, artista, artesano, artificial
Pan	panadero, panificar, panera, panadería
Jabon	jabonera, enjabonar, desenjabonar, jabonoso
Mar	marítimo, marino, altamar
Pintar	pintura, pintor, pintada

Las letras resaltadas en color rojo en la Tabla 2.6 es la secuencia de letras que no cambia en ningún término que componga la familia léxica. Las palabras que no están resaltadas en color rojo son los ya mencionados afijos o morfemas (prefijo, sufijo, etc.).

2.3. Recuperación de información

El gran número de hiperdocumentos y el incremento exponencial de información, aunado a la falta de estructuración lógica, da como resultado problemas para la Recuperación de Información (IR). De acuerdo a [Manning et al., 2008] la IR podría ser definida como:

"IR es encontrar material (usualmente documentos) de una naturaleza no estructurada (usualmente texto) que satisface una necesidad de información desde el interior de grandes colección (usualmente almacenadas en las computadoras)"

A grandes rasgo, la IR pretende extraer datos relevantes de un conjunto de elementos (documentos, páginas, etc.) de acuerdo a una determinada consulta. Para este fin surgieron modelos que permiten extraer estos datos de una manera precisa. En la sección 2.3.1 se muestra una jerarquización de los modelos existentes para la recuperación de información (IRM).

2.3.1. Taxonomía de Modelos de Recuperación de Información (IRM)

Los IRM clásicos que se observan en la Figura 2.3 se clasifican en tres diferentes tipos: teoría de conjuntos, algebraicos y probabilísticos [Baeza-Yates et al., 1999]. Estas clasificaciones se dan en base a la naturaleza que algunos modelos comparten, sin embargo los modelos pueden diferir tanto en su ejecución, como en la lógica de sus procesos, así como en los elementos que requieren.

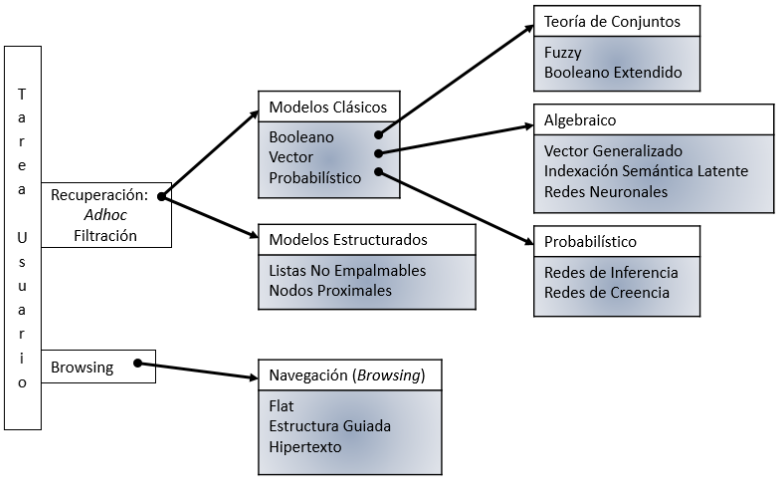

Figura 2.3: Clasificación de los IRM

Un modelo pertenece al tipo Teoría del Conjunto, si los documentos y las consultas son representados como conjuntos de términos del índice, como el Modelo Booleano. Así mismo se dice que un modelos es del tipo Algebraico si los documentos y las consultas son representadas en vectores en un espacio de n-dimensiones, tal como el Modelo Vectorial. Por ultimo un modelo es probabilístico si está basado en la teoría probabilística [Gudivada et al., 1997].

Aun en la diversidad de modelos para la IR, estos comparten una representación, caracterización o metodología general. [Baeza-Yates et al., 1999] define una caracterización en forma de un cuadruple:

$$[\mathbf{D}, \mathbf{Q}, \mathcal{F}, R(q_i, d_j)] \tag{2.1}$$

Donde $\mathbf{D}$ es un conjunto de vistas lógicas(o representaciones) para los documentos en la colección. $\mathbf{Q}$ es un conjunto compuesto de vistas lógicas (o representaciones) para la información que el usuario necesita, tales representaciones son llamadas consultas. $\mathcal{F}$ es un *framework* para modelar las representaciones de documentos, consultas y sus relaciones. $R(q_i, d_j)$ es una función de clasificación la cual asocia un número real con una consulta $q_i \in \mathbf{Q}$ y la representación del documento $d_j \in \mathbf{D}$. Tal clasificación define un ordenamiento entre los documentos con relación a la consulta q_i.

2.3.2. Colecciones estándar

En un IRS se requiere de una base documental para realizar las pruebas de clasificación, dicha base es conocida como Colección de Documentos. En esta sección, se abordara una breve reseña de la colección estándar CACM.

Colección estándar `CACM`

CACM es un conjunto de 3204 artículos científicos del 1958 a 1979 con dominio en

Ciencias de la Computación. Sin embargo cada archivo de la colección solo contiene información estructural como sigue:

- Nombre de autores
- Fecha de publicación
- Palabra en estado raíz del título y las secciones abstractas
- Referencias directas entre los artículos
- Conexiones de acoplamiento bibliográficas
- número de Co-citaciones por cada par de artículos

En general la colección está compuesta por una conjunto de documentos científicos con dominio en ciencias de la computación, un archivo *.txt* con 64 consultas de prueba y un archivo *.txt* con las referencias de los documentos relevantes para cada consulta.

También existen otras colecciones que están estrechamente relacionadas con CACM, tales colecciones se muestran en la Tabla 2.7. La primera columna indica el nombre de la Colección. La segunda columna hace referencia al dominio de la colección. La tercera columna indica el número de documentos que componen la colección. Finalmente en la cuarta columna indica el número de consultas de la colección [Baeza-Yates et al., 1999].

Tabla 2.7: Colección de documentos relacionados con CACM.

Colección	**Tema**	**Num. Doc.**	**Num. Cons.**
ADI	Ciencias de la información	82	35
CACM	Ciencias de la Computación	3204	64
ISI	Ciencia de la biblioteca	1460	76
CRAN	Aeronáutica	1400	225
LISA	Ciencia de la biblioteca	6004	35
MED	Medicina	1033	30
NLM	Medicina	3078	155
NPL	Ingeniería eléctrica	11,429	100
TIME	Artículos generales	423	83

2.3.3. Índice Inverso (*Inverted Index*)

De acuerdo a lo que se observó en la sección 2.3.2, las colecciones de documentos tienden a ser cuantiosas de información y en consecuencia empeoran el tiempo en el

proceso de búsqueda. Es por esto que un *Inverted File* o *Inverted Index* es un mecanismo de palabra orientada para indexar una colección de documentos en orden para acelerar la tarea de búsqueda [Baeza-Yates et al., 1999]. La estructura del *Inverted Index* se compone de dos elementos: el vocabulario y la referencia de los documentos. El vocabulario son todas las palabras que no sean *stopwords* que contenga la colección. La referencia de los documentos está ligada a la concurrencia de éstas palabra en el documento. En la Figura 2.4 se muestra un ejemplo de la manera en que se construye un *Inverted Index*.

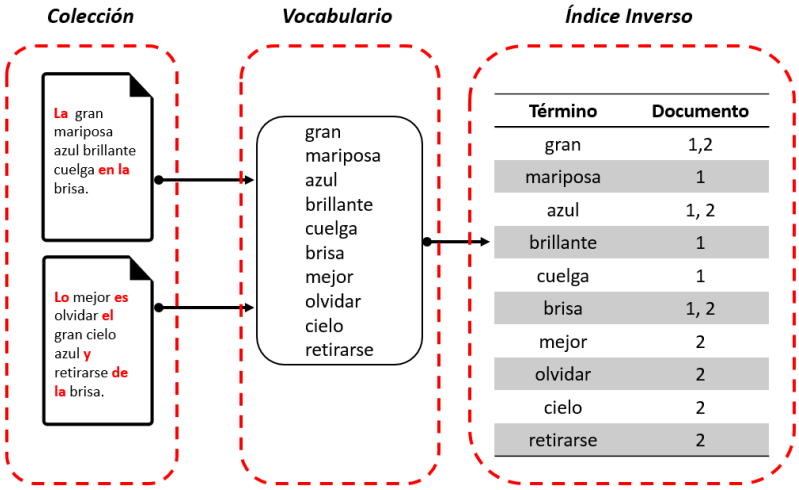

Figura 2.4: Construcción de un *Inverted Index*.

El medio más usual para guardar un *Inverted Index* es un archivo de texto plano. Dentro de este archivo se encuentran los dos elementos que componen al índice. El vocabulario son las palabras que aportan de información significativa para mejorar el proceso de selección de documentos, es decir palabras que no sean *stopwords*. A cada término que constituye al vocabulario se le asocia un conjunto de identificadores que hacen referencia a los documentos donde dicho término se manifiesta.

2.3.4. Índices de evaluación

Para determinar el nivel de eficacia de un IRS se evalúan los resultados obtenidos mediante algunas métricas de evaluación. Las dos más frecuentes y básicas métricas para la recuperación son *Precision* y *recall* [Manning et al., 2008].

Considere una colección de datos estándar y una solicitud de información (consulta). Sea $|R|$ el número de documentos relevantes para la consulta. Sea $|A|$ el número de documentos recuperados para la consulta. Y sea $|Ra|$ el número de documentos relevantes recuperados. Entonces se dice que *Precision* y *Recall* son [Baeza-Yates et al., 1999]:

Recall es la fracción de los documentos relevantes $\in |R|$ que han sido recuperados.

$$Recall = \frac{|Ra|}{|R|} \tag{2.2}$$

Precision es la fracción de los documentos recuperados $\in |A|$ que son relevantes.

$$Precision = \frac{|Ra|}{|A|} \tag{2.3}$$

Sin embargo la métrica *Precision* tiende a ser deficiente de acuerdo a su definición. Esto debido a que si el conjunto $|A|$ es en exceso mayor al conjunto $|Ra|$ el porcentaje de *Precision* decrecerá exorbitantemente. Es por esto que es vital determinar la precision de un documento relevante considerando la posición en donde fue colocado tras el proceso de clasificación.

La métrica que aplica lo anterior es $AvgP$. La precisión de cada documento relevante se adquiere dividiendo el número de documento relevante encontrado hasta el momento entre la pocisión donde se encuentre, ambos valores adquiridos del conjunto de documentos recuperados [Manning et al., 2008]. Entonces se puede decir que la precision promedio del conjunto de documentos relevantes recuperados está definido por la Ecuación 2.4. Por consiguiente la métrica MAP está constituida por la Ecuación 2.5.

$$AvgP = \frac{\sum_{i=1}^{n} P(i)rel(k)}{|Ra|} \tag{2.4}$$

$$MAP = \sum_{q=1}^{Q} AvgP(q) \tag{2.5}$$

2.3.5. Modelo de Espacio Vectorial (VSM)

VSM o Modelo Vectorial es uno de los modelos más utilizados dentro de IR, esto se debe a que en númerosos trabajos [Sedeño, 2011, La Serna Palomino et al., 2009, Montoya Pérez, 2012, Monsalve, 2012] han comprobado la eficiencia y sencillez del modelo sobre otros. De acuerdo a [Baeza-Yates et al., 1999] define al *VSM* como sigue:

Definición. Para el Modelo Vectorial, el peso $w_{i,j}$ asociado con un par (k_i, d_j) es positivo y no binario. Además los términos del indice en la consulta son solo ponderados. Sea $w_{i,j}$ el peso asociado con el par $[k_i, d_j]$, donde $w_{i,j} \geq 0$. El vector de la consulta $\vec{q}$ está definida como $\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$ donde t es el número total de términos indexados en el sistema. Como antes, el vector del documento d_j está representado por $\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$.

Un documento d_j y una consulta de usuario q son representadas como vectores con t–dimensiones. El propósito del VSM es evaluar el grado de similitud de un documento d_j y una consulta q mediante la correlación entre los vectores $\vec{d_j}$ y $\vec{q}$ [Baeza-Yates et al., 1999].

Para efectuar el proceso de similitud se tiene la Ecuación 2.6 conocida como *el coseno del algulo*, *coseno de similitud* o simplemente coseno. Así como la Ecuación 2.7 nombrada *Soft Cosine Measure* por [Sidorov et al., 2014].

$$coseno(q, d_1) = \frac{\sum_{i=1}^{n} P_i W_{ij}}{\sqrt{\sum_{i=1}^{n} P_i}\sqrt{\sum_{i=1}^{n} W_{ij}}} \tag{2.6}$$

$$soft\_cosine_1(a, b) = \frac{\sum \sum_{i,j}^{N} S_{ij} a_i b_j}{\sqrt{\sum \sum_{i,j}^{N} S_{ij} a_i a_j}\sqrt{\sum \sum_{i,j}^{N} S_{ij} b_i b_j}} \tag{2.7}$$

La diferencia de *Soft Cosine Measure* con respecto al coseno de similitud, está en que toma en cuenta las bases de los vectores, representadas en S_{ij}. Los valores que toman lugar en las Ecuaciones 2.6 y 2.7 se obtienen por medio de dos elementos:

- Matriz de pesos de la colección (*Weight Matrix*).
- Un vector de pesos de la consulta (*Weigth Query*).

La matriz de pesos normaliza los valores de una matriz de ocurrencia dando lugar a el peso de un término t_i en un documento d_j. La matriz de ocurrencia guarda la frecuencia del término t_i en un documento d_j. Para poder generar la matriz de pesos se emplea la ecuación 2.8. Para poder generar los pesos de cada término del vector consulta se requiere aplicar la ecuación 2.9.

$$w_{t,d} = tf\_idt_{t,d} = tf_{t,d} \times idf_{t,d} \tag{2.8}$$

$$P_{q_i} = \frac{tf_{q_i}}{max(tf_{q_i})} idf_{t,d} \tag{2.9}$$

$$idf_t = log(\frac{N}{df_t}) \tag{2.10}$$

Donde tf_{q_i} representa la frecuencia del término q_i en el vector de la consulta, $max(tf_{q_i})$ indica el valor de frecuencia máxima que se encuentra en la consulta, finalmente idf_t es el valor de la frecuencia inversa del término dado por el logaritmo del total de términos sobre la cantidad de documentos donde ese término aparece.

El peso de cada término se obtiene realizando el producto de la frecuencia del término con la frecuencia inversa de dicho término. Esta estructura, junto con el vector de pesos de la consulta, son las más cruciales debido a que permiten medir el grado de relevancia que tiene un documento para una consulta.

2.4. Ontología

Ontología es un término adquirido de la filosofía que refiere a la ciencia de la descripción de los tipos de entidades en el mundo y como están relacionados [W3C, 2004]. Pero en el ámbito de la informática existe una afirmación proporcionada por [Gruber, 1993] que describe de manera concisa a una ontología:

"Una ontología es una especificación explicita de una conceptualización"

Es decir, permite obtener un vasto vocabulario del conocimiento de uno o varios dominios representados en un esquema regido por un conjunto de reglas y axiomas de inferencia para deducir nuevo conocimiento, sin dependencia sobre el propósito de su uso, para consumar un trabajo en específico. Posteriormente se extendió esta definición por [Benjamins et al., 1998] :

"Una ontología es una especificación explicita y formal de una conceptualización"

Pero la primera definición con respecto a las ontologías en el ámbito de la informática, la proporciono [Neches et al., 1991] que dice:

"Una ontología define los términos y las relaciones básicas que componen el vocabulario de una área temática, así como las reglas para combinar términos y relaciones para definir extensiones al vocabulario"

Las ontologías son una herramienta ideal para representar conocimiento de una forma sencilla tanto para el ser humano como para una máquina. Estas permiten mediante su logística ser construidas de una manera intuitiva, con el fin de representar claramente el conocimiento de algún tema de interés de manera conceptual.

2.4.1. Componentes

La estructura de una ontología contiene una serie de elementos que permiten construir el modelo conceptual de un cierto dominio. A continuación se muestran los elementos básicos que una ontología adquiere de acuerdo a [Gruber, 1993].

- Conceptos. Son las ideas básicas que intentan formalizar, estos conceptos puedes ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- Relaciones. Representan la interacción y el enlace entre los conceptos del dominio.
- Funciones. Son un tipo concreto de relación, donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- Axiomas. Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si X y Y son de la clase Z, entonces X, no es subclase de Y" o "Para todo X que cumpla con la condición Cond1, A es B", etc.
- Instancias. Representan objetos determinados de un concepto.

El esquema de una ontología puede parecer una serie de conceptos enlazados y con un orden jerárquico, con ciertos criterios o reglas para definir aún mejor una representación de conocimiento. Esta serie de componentes son los que puede contener toda ontología.

2.4.2. Tipos y Clasificación

Debido a la variedad para lo que una ontología puede ser usada y dependiendo su propósito, se pueden mencionar diferentes clasificaciones. Dichas categorías son propuestas por [Guarino, 1999] y [Van Heijst et al., 1997]:

- Dependiendo de la magnitud de conocimiento representado.
  - Ontologías de alto nivel o genéricas. Son ontologías que describen los conocimientos generales, se compone de una colección de ontologías de dominio.
  - Ontologías de dominio. Es un vocabulario, donde representan el conocimiento de un dominio en concreto.
  - Ontologías específicas, tareas o técnicas. Describen una tarea, actividad o artefacto, por ejemplo componentes, procesos o funciones.
  - Ontologías de aplicación. Describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.
- Tipo de agente dirigida
  - Ontologías lingüísticas. Diseñado con aspectos gramáticos, semánticos y sintácticos para el entendimiento del ser humano.
  - Ontologías no lingüísticas. Diseñados para el entendimiento de agentes inteligentes o robots.
  - Ontologías mixtas.
- Dependiendo del grado de abstracción y razonamiento lógico
  - Ontologías descriptivas. Incluyen descripciones, taxonomías de conceptos, relaciones entre los conceptos y propiedades, pero no permiten inferencias lógicas.
  - Ontologías lógicas. Permiten inferencias lógicas mediante la utilización de una serie de componentes como la inclusión de axiomas.

Esta serie de clasificaciones, difieren de acuerdo al punto de vista en que se analiza una ontología, por ende no son solo las únicas clasificaciones que podrían surgir. Las anteriores características engloban de manera general como diferenciar una ontología.

2.4.3. Lenguajes de Ontología

Dentro de los lenguajes que han surgido para la creación de ontologías, se encuentran SHOE (Simple HTML Ontology Extensions), OIL (Ontology Inference Layer o

Ontology Interchange Language), DAML (DARPA Agent Markup Language), OWL (Web Ontology Language), RDF (Resource Description Framework), RDF-S (Resource Description Framework Schema), entre otras.

El primer lenguaje fue pionero del lenguaje de etiquetado para diseño de ontologías, SHOE (desarrollado por Sean Luke, Lee Spector, James Hendler, Jeff Helfin y davir Rager en 1996) es un lenguaje de representación de conocimiento basado en HTML, siendo un reducido conjunto de éste, el cual agrega las etiquetas necesarias para incrustar arbitrariamente datos semánticos en las páginas web [Heflin et al., 1997].

Para establecer un lenguaje con nuevas propiedades y estructuras que permitieran definir y mejorara el modelado y entendimiento de una ontología, algunos lenguajes incorporaron propiedades de otros lenguajes, tal es el caso de DAM, que se unió a OIL creando así DAM + OIL. Aun con estas adaptaciones, en la actualidad los lenguajes que encabezan las listas de popularidad son RDF y OWL.

Aun cuando existe una cierta variedad de lenguajes para ontologías, los más colectivos son RDF y OWL. En donde RDF es un modelo estándar para el intercambio de datos en la Web teniendo características que facilitan la fusión de datos, incluso si los esquemas subyacentes difieren, y soporta específicamente la evolución de esquemas en el tiempo sin necesidad de todos los consumidores de datos que pueden cambiar [W3, 2014].

RDF está basado en la idea de identificar los recursos en la web usando URL y describiendo los recursos en términos de propiedades simples y valores. Una descripción de éste es un conjunto de proposiciones simples conocidas como tripletas, a causa de su composición: sujeto (recurso), predicado (propiedad) y objeto (valor de la propiedad).

Las tripletas puedan ser representadas en nodos conectadas por líneas con etiquetas, donde los nodos son los recursos y las líneas son las propiedades de esos recursos. Un ejemplo de una tripleta RDF se muestra en la Figura 2.5.

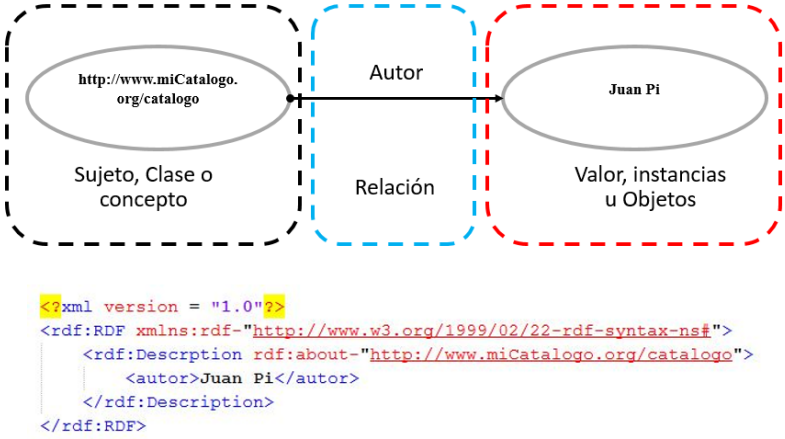

Figura 2.5: Tripleta en RDF: Grafo y representación en XML.

Mientras tanto, OWL está diseñado para el uso en aplicaciones que necesitan procesar el contenido de información en lugar de sólo presentar la información a los seres

humanos. Por lo tanto permite una interpretación a nivel máquina del contenido web, que con ayuda de XML, RDF y RDF- S proporciona un vocabulario adicional junto con una semántica formal [W3, 2014].

Capítulo 3

Estado del arte

En este apartado, se redacta un resumen de los proyectos que involucran la utilización de Modelos de Recuperación de Información suscitados en los últimos años. Así mismo, se presentan trabajos relacionados con la expansión de consultas utilizando ontologías. Al término de cada sección (3.1 y 3.2) se muestra una tabla general de los proyectos de estos.

3.1. Modelos de Recuperación de Información

En el trabajo [Farrús and Costa-jussà, 2013] analiza y discute algunos de los más recientes sistemas de evaluación educativa en línea, dando como caso de uso la Universidad Abierta de Cataluña. Este sitio emplea el IRM de Indexación Semántica Latente (LSI) para la evaluación de ensayos y preguntas, permitiendo al usuario auto-evaluarse y recibir retroalimentación en cualquier momento y de forma inmediata de los exámenes y ensayos que envía. Para determinar si el ensayo o las respuestas de los exámenes son correctos se utiliza la métrica de similitud de LSI, la cual permite obtener el porcentaje de similitud entre las respuestas que envía el estudiante y la solución que proporciona el profesor. El índice utilizado fue *Precision*, el cual mide el porcentaje de documentos relevantes extraídos entre el total de documentos recuperados. Los resultados arrojaron que el mayor porcentaje de *Precision* obtenido fue del 50 %, debido a que dichos exámenes y ensayos tienen opción de contestarlos en catalán como español, y los resultados correctos están solo en catalán, lo cual disminuye sustancialmente este índice.

Así como hay sistemas que evalúan la similitud entre documentos, existen otros que evalúan y recuperan imágenes. Tal es el caso de [la Serna Palomino et al., 2013], que diseñaron un IRS de imágenes faciales, empleando VSM. Este sistema recupera las 3 imágenes con el mayor grado de similitud con las características ingresadas por el usuario. Este sistema consta de dos módulos: agregación de imágenes con sus perfiles y la búsqueda de una imagen dada una consulta. Las imágenes al momento de ser agregadas al sistema, se les vincula con un perfil, el cual contiene una serie de características. Las características pueden ser faciales (en tamaño, color, forma, etc.) y/o descripción textual. La métrica utilizada fue el coseno de similitud y el índice fue *Precision.*[FALTA: porcentaje precisión]

Dentro de los modelos de RI, se encuentran métricas para obtener el grado de similitud, tales como Jaccard, DICE y el coseno de similitud, entre otros. En el trabajo realizado por [Monsalve, 2012] realiza pruebas sobre, cuál de las tres métricas mencionadas con anterioridad, proporciona mejores resultados tanto en *Accuracy* (precisión) como en *Recall* (desempeño). Para realizar el experimento se empleó el VSM y la colección estándar ADI. Este experimento arrojó que la mejor precisión se obtuvo al utilizar la métrica Jaccard.

Normalmente, las consultas enviadas al IRM están compuestas por términos (palabras) que aporten información sustancial al modelo. Pero en el caso de [Montoya Pérez, 2012], emplea una nueva técnica llamada *Términos Latentes* un poco similar a la técnica *n-gramas*. Esta nueva técnica agrupa uno o más términos para tomar esta nueva agrupación como un nuevo nodo, generando una especie de estructura de árbol. Esto quiere decir, que no sólo crea un vocabulario de términos, si no también integra frases que van compuestas desde dos hasta el número máximo de términos que contenga la consulta. Para esta evaluación, se utilizó el Modelo Vectorial, también conocido como Modelo de Espacio Vectorial. Como documentos de prueba fueron utilizadas la colección llamada la Biblioteca del CIC. La calidad de su propuesta fue evaluada utilizando las métricas de Precisión, Cobertura y Exactitud con la calidad del Modelo Vectorial Clásico. Sus resultados indican que evaluar frases mejora la precisión que utilizando solo términos, como lo hace el Modelo Vectorial Clásico.

En el trabajo de [Sedeño, 2011] desarrollo un SRI para clasificar y agrupar textos de noticias en dos fases. La primera fase consiste en la agrupación de textos, utilizando el algoritmo de *Extended Star Clustering*, con la finalidad de estructurar mejor la información en la base de datos, y así agilizar el proceso de búsqueda, ya que considero que es más eficiente acceder a un documento en una colección de datos agrupada con ciertas características que en una colección con documentos dispersos. En la segunda fase, es utilizado el Modelo Vectorial para clasificar las noticias más relevantes. Desafortunadamente reportó que los *clusters* no tenían una buena calidad, debido a que la función de similitud no era la más apropiada, porque no incluye rasgos descriptivos de los documentos.

En el trabajo de [La Serna Palomino et al., 2009] implementaron un SRI capaz de trabajar con documentos digitales de diferentes dominios, el cual está compuesto por dos interfaces. La primera interfaz está encargada de realizar consultas para extraer documentos relevantes. Para el proceso de clasificación se utilizó el Modelo de Espacio Vectorial y los experimentos fueron evaluados con las métricas de *Accuracy* y *Recall*, sin reportar dichos valores. La segunda interface consta de un formulario que permite registrar y cargar un nuevo documento, incrementando de esta manera su colección de documentos.

En el trabajo de [Ding et al., 2006], se realizaron pruebas de la combinación del modelo de Indexación Semántica Latente Probabilística (ISLP) y la técnica de Factorización de la Matriz no Negativa (FMN). La experimentación consistió en comparar los modelos originales contra un modelo compuesto de estos. En la experimentación reportaron los índices de *Accuracy* y *Recall*. Utilizaron las colecciones estándares CSTR, WebKB, Log, Reuters y WebAce. La experimentación con los modelos originales, no percibió una diferencia significativa. Sin embargo el modelo híbrido proporciono mejores índices que el de los modelos originales.

En la Tabla 3.1 se muestra un resumen de los trabajos mencionados con anterioridad. En la primera columna se plantea el nombre del trabajo. La segunda columna corresponde al Modelo empleado. La tercera columna indica cuales fueron los índices de evaluación utilizados. En la cuarta columna se proporciona el nombre de la o las colecciones utilizadas. En la última columna se da una breve descripción del trabajo.

Tabla 3.1: Características principales de los trabajos relacionados con IRM.

Referencias	**Modelos**	**Índices de evaluación**	**Colecciones**	**Características**
Presencia de IRRODL Evaluación Automática [**Farrús and Costa-jussà, 2013**]	Análisis semántico latente o Indexación Semántica Latente	Precisión	Tareas digitales de la Universidad abierta de Cataluña.	-Sistema de educación, evaluando textos digitales sobre tareas matemáticas. -Plataforma de ingeniería en español y catalán. -Evalúa y califica las tareas. -Compara resultado automático con manual.
Diseño de un IRS de imagines de individuos malhechores para Seguridad Ciudadana [**la Serna Palomino et al., 2013**]	Modelo de Espacio Vectorial		Información e imágenes de criminales	-Es en lenguaje natural: información, contenido o ambas. -El SRI consta de dos módulos: almacenamiento y recuperación. -Se realiza el CS con las características. -Se ordenan las imágenes por orden de similitud.
Construcción de un Árbol de TL y su uso en Semejanza de Doc. [**Montoya Pérez, 2012**]	Modelo de Espacio Vectorial	Precisión / Cobertura / Exactitud	Biblioteca del CIC	-Compara el VM y el VSM. -Utiliza palabras y frases para la evaluación. -Emplea el modelo de árbol de términos latentes.
Clasificación y agrupamiento de textos de noticias. [**Sedeño, 2011**]	Modelo de Espacio Vectorial	Precisión	-	-Utiliza el algoritmo Extended Star Clustering para la agrupación. -Evalúa la similitud de los documentos, sin reportar buenos resultados.
Implementación de un IRS [**La Serna Palomino et al., 2009**]	Modelo de Espacio Vectorial	Precisión /Performance	General	-Carga manualmente los documentos electrónicos. -Contiene documentos en diferentes formatos de diferentes áreas y temas.
NMF and PLSI: equivalence, Chi-square and HM. [**Ding et al., 2006**]	Probabilistic Latent Semantic Indexing / Non- negative Matrix Factorization	Accuracy / Recall	CSTR / WebKB / Log /Reuters / WebAce	-Realiza comparativas entre LSI and NMF. -Realiza comparativas entre LSI and algoritmo hibrido.
Experimento de RI usando métricas CS, Jaccard y DICE. [**Monsalve, 2012**]	Modelo de Espacio Vectorial	Precisión / Recall	ADI	-Compara las métricas del Coseno, Jaccard y DICE.

3.2. Expansión de Consultas con ontologías

Dentro de los trabajos reportados en los últimos años sobre la expansión de consultas utilizando ontologías, se encuentran los siguientes:

En el trabajo de [Valbuena and Londoño, 2014] presentaron un sistema de búsqueda con el uso de un sistema de indexación ontológica aplicando una técnica de emparejamiento de retículos con dominio biomédico. Las consultas se procesan con asociación de términos ontológicos para construir el retículo. Se realiza un emparejamiento o comparación del retículo de la consulta con el retículo de la colección de documentos nivel por nivel, extrayendo aquellos nodos que coincidan entre sí. El retículo de la colección de documentos es el conjunto de cada árbol conceptual generado en cada documento. Posteriormente la métrica de coseno es aplicada para clasificar los documentos más importantes, de acuerdo al proceso de emparejamiento. Al final aplica pruebas de precisión, exhaustividad y medida F para evaluar sus resultados.

Otro de los trabajos que aplica la expansión de consultas utilizando ontologías es el trabajo de Kuna et al [Kuna et al., 2014], donde propone un meta-buscador de artículos científicos en el área de ciencias de la computación, donde aplica dos ontologías, que conceptualmente son la misma pero de diferente idioma (específicamente inglés y castellano). El procesamiento de la consulta se puede listar como sigue: 1. Identifica el idioma de la consulta con una instancia auxiliar para seleccionar la ontología pertinente; 2. Busca los términos de la consulta en la ontología y los etiqueta como "términos_candidatos"; 3. A los "términos_candidatos" se le extraen el concepto "padre", los conceptos hermanos (términos del mismo nivel) y los sinónimos, así mismo se realiza el proceso de traducción, que implica convertir los "términos_candidatos" al idioma inverso; 4. Se generan las 4 expansiones o concatenaciones de la consulta original en lenguaje original e inverso, con las siguientes reglas lógicas: consulta original AND término candidato, término candidato AND concepto padre, término candidato AND concepto hermano, término candidato AND concepto sinónimo. Una vez obtenidas las expansiones de la consulta original, se procede a clasificar los documentos, tomando en cuenta los parámetros FP (fuentes de publicación), A (autores) y D (documentos) para complementar la métrica Q, que es su medida de similitud.

Dentro de los trabajos que expanden las consultas con el fin de mejorar las métricas de *Accuracy* y *Recall*, está el trabajo de [Pabón et al., 2014] que radica en la ejecución de dos técnicas de IR: el modelo de espacio vectorial y las anotaciones semánticas. La segunda utiliza una ontología con dominio de Ciencias de la Computación propuesta por ACM que puede procesar lenguaje en inglés y español. La colección de documentos que emplean, entra en un pre-procesamiento inicial, en donde se incorpora a la estructura conceptual de la ontología, es decir, mediante la API GATE vincula cada documento con la ontología, permitiendo que este documento se coloque en la clase que más concurrencia tenga de este documento. En el mecanismo, está como principio el procesamiento de la consulta, donde aplica técnicas de procesamiento de lenguaje natural para destacar las palabras claves de la misma. Después las palabras clave entran en un procesamiento paralelo entre las dos técnicas de IR aplicadas, al final de la ejecución de cada técnica, se aplica un mecanismo de clasificación híbrido. Este mecanismo combina los índices proporcionados por cada técnica para mostrar al usuario los documentos relevantes, y estos dependerán de dos factores entre el 0 y 1: ω para las

anotaciones semánticas y λ para el modelo de espacio vectorial, siguiendo las siguientes dos configuraciones: ω=0.7 y λ=0.3 si los consulta se representa completamente con la ontología, ω=0.4 y λ=0.6 si la consulta no se representa en su totalidad en la ontología.

En la Tabla 3.2 se muestra un resumen sobre las características más relevantes de los trabajos mencionados con anterioridad. En la primera columna se plantea el nombre del trabajo. La segunda columna corresponde al Modelo empleado. La tercera columna indica cuales fueron los índices de evaluación utilizados. En la cuarta columna se proporciona el nombre de la o las colecciones utilizadas. En la última columna se da una breve descripción del trabajo.

Tabla 3.2: Características principales de los trabajos relacionados con expansión de consultas.

Referencias	**Modelos**	**Métricas**	**Colecciones**	**Características**
Desarrollo de un SRI para Publicaciones Científicas del Área de Ciencias de la Computación [**Kuna et al., 2014**]	Modelo propuesto:Meta-buscador	*Accurracy*, Calidad de la fuente, de autores y del artículo	Google Scholar, ACM, IEEE Xplore	-utiliza ontología para expansión de consultas - la ontología es del dominio en sub-área en IA - utiliza métricas para evaluación de artículos, tales como: Factor de impacto, SJR, Ranking Core, Índice H, G, AR y Cantidad de Citas.
Propuesta para extender semánticamente el proceso de RI [**Pabón et al., 2014**]	Modelo de Espacio Vectorial	*Accurracy*, *Recall*	Biblioteca Digital ACM	-Utiliza ontología de multi-leguaje para procesar la colección de docentes. -El SRI consta de dos módulos: MEV y Anotaciones semánticas. -Utiliza una métrica de similitud, propuesta por Fernández y Vallet 2007.
Modelo propuesto	Modelo de Espacio Vectorial	Precisión / Exactitud	CACM	-Expansión de Consultas utilizando una ontología. -La ontología contendrá un vocabulario sobre el área de Ciencias de la Computación. -La evaluación se limita a la similitud de documentos.

En la última fila de la Tabla 3.2 se dan las características respectivas a éste trabajo. Dando notoriedad a la utilización de una ontología para expandir las consultas de la colección CACM utilizando el VSM.

Capítulo 4

Diseño e implementación

Para este capítulo se tiene contemplada una serie de pautas para el diseño e implementación del VSM (Modelo de Espacio Vectorial) y la ontología con dominio en Ciencias de la Computación, así como todos aquellos requerimientos necesarios para su construcción. Para esto, se tienen las siguientes secciones para abordar con mayor detalle cada pauta.

Como principio se aborda la Sección 4.1, la cual muestra la metodología aplicada para la obtención del *Inverted Index* y un conjunto de archivos con características representativas de cada documento de la colección. En la Sección 4.2 de igual forma se observa la metodología aplicada para la construcción de este modelo, así mismo la ontología en la Seccion 4.3. Por último se muestra la ontología y descripción de las diferentes expansiones aplicadas a las consultas.

4.1. Pre-procesamiento inicial

Los IRM requieren iniciar con dos factores para poder emprender el proceso de Recuperación de Información. El primer factor es una colección de documentos que sirvan como una base de datos para la búsqueda y clasificación de información. El segundo factor es un índice que provee al modelo la eficiencia necesaria para realizar las búsquedas pertinentes utilizando los términos concurrentes de la colección, conocida como *Inverted Index*.

El *Inverted Index* como se explica en la Sección 2.3.3 permite agilizar la búsqueda de documentos que podrían ser relevantes para una consulta tomando como referencia todos los términos de la colección de documentos. Todos los términos clave que componen a la consulta son buscados en dicho índice permitiendo extraer la referencia de aquellos documentos en donde dichos términos concurren. Basta decir que la colección es fundamental para poder construir este *Inverted Index*.

El *Inverted Index* provee de una estructura eficiente para referenciar todos aquellos documentos que contienen un término en particular. En la Figura 4.1 se puede apreciar la estructura del *Inverted Index* implementado en este trabajo.

$$\begin{bmatrix} t_{1c} & [d_1, d_2, \dots] \\ t_{2c} & [d_3, d_5, \dots] \\ t_{3c} & [d_1] \\ t_{4c} & [d_{10}, d_7, \dots] \\ t_{5c} & [d_9, d_{11}] \\ \vdots & \vdots \\ t_{nc} & [d_3, d_5, \dots] \end{bmatrix}$$

Figura 4.1: Estructura del *Inverted Index*.

Como se aprecia en la Figura 4.1, el *Inverted Index* tiene dos columnas que corresponden a los términos de la colección t_{nc} y a los vectores de documentos que contienen la referencia de los documentos donde dicho término concurre al menos una vez. Estos vectores permiten saber que documentos formaran parte del conjunto de documentos recuperados. En la Figura 4.2 se observa un fragmento (15 líneas) de la composición del *Inverted Index*, debido a que el número de términos del índice haciende a 31,705.

```
1  program ---> [1, 2, 3, 5, 7, 8, 9, 12, 14, 15, 17, 19, 20, 21,
2  processor ---> [1, 17, 39, 41, 44, 55, 70, 76, 83, 87, 88, 95,
3  time ---> [1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 14, 15, 17, 18, 19,
4  memory ---> [1, 2, 6, 7, 9, 10, 14, 15, 23, 26, 29, 35, 37, 39,
5  may ---> [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 
6  Can ---> [1, 15, 34, 106, 112, 129]
7  instructions ---> [1, 7, 8, 9, 10, 13, 14, 18, 19, 26, 29, 31,
8  supervisor ---> [1, 99, 111, 115, 121, 122, 130]
9  Are ---> [1, 39, 43, 112]
10 processing ---> [1, 8, 14, 15, 19, 36, 37, 39, 44, 46, 50, 55,
11 statements ---> [1, 3, 7, 13, 14, 15, 17, 18, 19, 22, 35, 36, 
12 The ---> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1
13 running ---> [1, 2, 11, 12, 14, 15, 20, 43, 50, 52, 62, 74, 78,
14 Is ---> [1, 6, 17, 39, 41, 64, 77, 90, 101, 109, 117, 121, 126,
15 data ---> [1, 8, 9, 14, 15, 18, 21, 31, 32, 36, 37, 38, 39, 40,
```

Figura 4.2: Pequeña sección del archivo *Inverted Index*.

Para poder adquirir la frecuencia de los términos en cada documento, se procesaron cada uno de los documentos que componen la colección. El formato de la Figura 4.3 hace referencia a las características obtenidas en este proceso. Donde en la primera columna aparece el identificador del documento. En la segunda columna aparece la localización del documento. Finalmente en la última columna aparece el conjunto de los términos que tiene el documento con sus respectivas frecuencias.

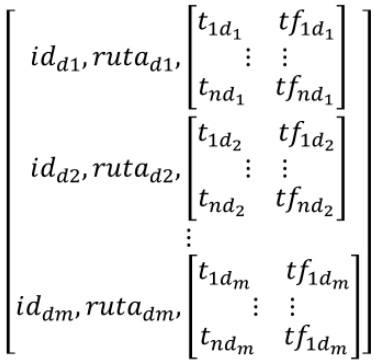

Figura 4.3: Información relevante de la estructura *Document*.

Para poder apreciar con mayor detalle el formato que cada documento adquirió tras la adquisición de su información se muestra un ejemplo en la Figura 4.4 del artículo titulado *A checklist of intelligence for programming systems*.

```
1  1
2  A checklist of intelligence for programming systems.pdf
3  0
4  program, 61
5  processor, 30
6  time, 28
7  memory, 25
8  may, 24
9  Can, 22
10 supervisor, 17
11 Are, 17
12 processing, 17
13 instructions, 17
14 Is, 16
15 running, 16
16 statements, 16
17 The, 16
18 data, 15
19 1, 14
20 used, 12
```

Figura 4.4: Ejemplo del formato que cada documento de la colección adquirió.

En el primer renglón del archivo de la Figura 4.4 se encuentra alojado el identificador del documento, que como se puede apreciar en este caso es el documento 1. En el segundo renglón se encuentra el nombre del artículo procesado. A partir del cuarto renglón comienza el conjunto de parejas de término - frecuencia. Cabe decir que la frecuencia radica sobre el número de veces que dicho término apareció en el documento. Por ejemplo, de acuerdo a la imagen el término *program* de la línea 4 aparece 61 veces en el documento.

Una vez que se obtuvo la información necesaria de la colección de documentos para crear el *Inverted Index* y el conjunto de documentos con el formato de la Figura 4.4 se precedió al desarrollo del VSM y los elementos que lo implican.

4.2. Modelo de Espacio Vectorial

Lo primero que se procedió a implementar fue el VSM. Su proceso se realizó por medio de varios métodos que procesan dos factores importantes: la colección de documentos y la consulta. Dichos métodos se pueden apreciar en el Algoritmo 1.

Algoritmo 1 Modelo de Espacio Vectorial

```
1:  procedure Modelo de Espacio Vectorial(Consultas)
2:      for consulta ∈ Consultas do
3:          leerIndiceInvertido()
4:          obtenerDocumentosRelevantes(consulta)
5:          generarMatrizDefrecuencia(consulta)
6:          generarFrecuenciaInversaDelDocumento(consulta)
7:          generarMatrizDePesos(consulta)
8:          generarPesoDeLaConsulta(consulta)
9:          generarMetricaCoseno(consulta)
10:     end for
11: end procedure
```

Como se observa en la linea 9 del Algoritmo 1, la métrica que usualmente se utiliza en este modelo es el coseno de similitud. Sin embargo, tras una experimentación que midió la precisión entre las métricas del coseno y *soft cosine measure* [Sidorov et al., 2014], se concluyó que la métrica *soft cosine measure* fue mejor y por ende fue utilizada. Dichos resultados se pueden apreciar en el Capítulo 5.

Uno de los métodos básicos del VSM es el que permite obtener la matriz de frecuencia. Esta matriz aloja la frecuencia que tiene cada término del *Inverted index* con los documentos candidatos. Los documentos candidatos son aquellos que comparten términos con la consulta. El Algoritmo 2 muestra el procedimiento implementado para conseguir dicha matriz.

Algoritmo 2 Generación de Matriz de frecuencia.

```
1:  procedure Generar Matriz de frecuencia
2:      for documento ∈ TodosDocumentos do
3:          ListaFrecuencia[ ] = frecuenciaTérminos(documento)
4:          for término ∈ ListaFrecuencia do
5:              for indice ∈ IndiceInvertido do
6:                  if término == IndiceInvertido[indice][0] then
7:                      setMatrizDefrecuencia(término, ListaFrecuencia[indice])
8:                  end if
9:              end for
10:         end for
11:     end for
12: end procedure
```

Como los IRM manejan grandes volúmenes de información es predecible decir que las frecuencias alojadas en la matriz de frecuencia tendrán valores exorbitantes. Para poder normalizar estos valores se procede a transformar estas frecuencias mediante *la*

frecuencia inversa del documento (Ecuación 2.10) representada por idf_t. Esta permite conocer la relevancia de un término t_i en una colección de datos D. El procedimiento implementado para obtener estos valores se observa en el Algoritmo 3.

Algoritmo 3 Generación de Frecuencia Inversa del Documento

```
1: procedure Generación de Frecuencia Inversa del Documento
2:     for indice ∈ FrecuenciaInversaDocumento do
       frecuenciaDocumento = matrizDefrecuencia[indice]
       contador = 0;
3:         for documento ∈ frecuenciaDocumento do
4:             if documento ≠ 0 then
5:                 contador++;
6:             end if
7:         end for
8:         if contador ≠ 0 then
9:             FrecuenciaInversaDocumento[indice] = log2(TamañoColección / contador)
10:        end if
11:    end for
12: end procedure
```

Una vez que se obtienen la matriz de frecuencias y el vector de la frecuencia inversa del documento se puede proceder a la obtención de la matriz de pesos. Esta matriz guarda la normalización de los valores de la matriz de frecuencia por medio del vector de la frecuencia inversa del documento. La ecuación 2.8 permite obtener lo dicho con anterioridad. En el Algoritmo 4 se muestra el procedimiento implementado para la obtención de la matriz de pesos.

Algoritmo 4 Generación de Matriz de Pesos

```
1: procedure Generación de Matriz de Pesos
2:     MatrizPesos[TotalDocumentos][Terminos]
3:     for documento ∈ MatrizPesos[TotalDocumentos] do
4:         for indice ∈ MatrizPesos[Terminos] do
5:             MatrizPesos[documento][indice] = matrizfrecuencia[documento][indice] *
   FrecuenciaInvertida[documento]
6:         end for
7:     end for
8: end procedure
```

La matriz de pesos es un factor crucial para el proceso de clasificación. Así mismo lo es el vector de pesos de la consulta, el cual por medio de la Ecuación 2.9 permite normalizar la frecuencia de los términos de una consulta.

Así como la matriz de pesos requirió de una matriz de frecuencia, el vector de pesos de la consulta necesito de un vector de frecuencia de la misma. Sin embargo el vector de frecuencia fue calculado dentro del método que obtiene el peso de este. Lo anterior se puede observar en las líneas de la 5 a la 11 del Algoritmo 5. Y a partir de la línea 12 se realiza el proceso de obtención del vector de pesos de la consulta.

Algoritmo 5 Generación de Pesos de la Consulta

procedure GENERACIÓN DE PESOS DE LA CONSULTA
    pesosConsulta[TerminosIndiceInverso]
    FrecuenciaTerminosConsulta[IndiceInverso][2]
    FrecuenciaTerminos[TerminosDeConsulta]
    for $\text{termino}_i \in$ ListaTerminosConsulta **do**
        for $\text{termino}_j \in$ ListaTerminosConsulta **do**
            if $\text{termino}_i = \text{termino}_j$ **then**
                FrecuenciaTerminos[termino_i]++
            end if
        end for
    end for
    for termino $\in$ ListaTerminos **do**
        for indice $\in$ IndiceInvertido **do**
            if IndiceInvertido[indice][0] = ListaTerminos[termino] **then**
                FrecuenciaTerminosConsulta[termino][0] = indice
                FrecuenciaTerminosConsulta[termino][1] = FrecuenciaTerminos[termino]
                pesosConsulta[indice] = $\frac{FrecuenciaTerminos[termino]}{\max FrecuenciaTerminos}$ * frecuenciaIndiceInvertidoDocumento[indice]
            end if
        end for
    end for
end procedure

Para poder proceder a clasificar los documentos relevantes de una consulta se utilizó la métrica del coseno suave [Sidorov et al., 2014]. Esta requiere de 4 factores cruciales: la matriz de pesos, la matriz de frecuencia, el vector de frecuencia y el vector de pesos. También se implementó la métrica del coseno (Algoritmo 6), pero la métrica del coseno suave arrojo mejores resultados en precisión que esta métrica.

Algoritmo 6 Generar lista de la métrica del Coseno de similitud.

procedure GENERAR COSENO DE SIMILITUD
    numerador = 0.0
    $\text{denominador}_c = 0.0$ ▷ Numerador de la consulta
    $\text{denominador}_d = 0.0$ ▷ Numerador del documento
    $\text{Lista}_{cos}[]$
    for doc $\in Doc_{rel}$ **do** ▷ Documentos Relevantes
        $\text{MetricaCoseno}_d = 0.0$ ▷ Calculo por documento
        for indice $\in pesosConsulta$ **do**
            numerador + = pesosConsulta[indice] * matrizPesos[indice][doc]
            $\text{denominador}_c += pesosConsulta[indice]^2$
            $\text{denominador}_d += matrizPesos[indice][doc]^2$
        end for
        $\text{MetricaCoseno}_d = \frac{numerador}{\sqrt{denominador_c} * \sqrt{denominador_d}}$
        $Lista_{cos}.add(MetricaCoseno_d, doc)$
    end for
end procedure

La diferencia de la métrica del coseno suave a la métrica del coseno es que utiliza los vectores base de cada término. Estos vectores están compuestos por la frecuencia del término. En los Algoritmos 7 y 8 se puede observar cómo se obtuvieron los valores S_{ij}, S_{jj} y S_{ii}. Estas variables son encontrados en el coseno suave (ecuación 2.7).

Algoritmo 7 Generar Coseno en S_{ii} y S_{jj} para Coseno Suave.

1: **procedure** GENERAR COSENO DE SIMILITUD($Frecuencia_i$)
2: $$\text{coseno} = \frac{Frecuencia_i^2}{\sqrt{Frecuencia_i^2} * \sqrt{Frecuencia_i^2}}$$
3: **end procedure**

Los factores que involucra el método del coseno suave son: la frecuencia de los términos de la consulta, los pesos del vector de consulta, la matriz de frecuencia y la matriz de pesos. En el Algoritmo 9 se puede observar el procedimiento que permitió calcular la similitud de los documentos con respecto a la consulta.

Algoritmo 8 Generar Coseno en S_{ij} para Coseno Suave.

1: **procedure** GENERAR COSENO DE SIMILITUD($Frecuencia_t, Frecuencia_d$)
2: $$\text{coseno} = \frac{Frecuencia_i^2}{\sqrt{Frecuencia_i^2} * \sqrt{Frecuencia_i^2}}$$
3: **end procedure**

Para aplicar el coseno suave localizado en la l+inea 33 del Algoritmo 9 se calcularon primeramente los valores de los factores S_{ij}, S_{ii} y S_{jj}. Los valores que adquieren estos factores son por medio de la métrica del coseno, pero en esta se sustituyen valores de frecuencia y no de peso como normalmente se haría en el coseno de similitud. Es decir, los factores reciben la frecuencia de un término $t_i \in FrecuenciaTerminosConsulta$ y/o reciben la frecuencia de un término $t_i \in MatrizDefrecuencia$. Lo anterior se puede observar en las lineas 13, 14 y 26 del Algoritmo 9.

De acuerdo a [Sidorov et al., 2014] cuando estos factores toman valores iguales a cero, se les asigna el valor de uno para regresar a la métrica del coseno suave a su estado natural, es decir, al coseno del ángulo. Esto con el fin de no alterar los resultados al realizar una operación matemática con factores iguales a cero. Lo anterior se puede apreciar en las líneas 16, 17, 18 y 28 del Algoritmo 9.

Algoritmo 9 Generar lista de coseno suave.

procedure GENERAR LISTA DE COSENO SUAVE.(Doc_{rel})
  $L_{cos} = [\dots]$
  $n = 0$ ▷ Numerador.
  $d_q = 0$ ▷ Denominador consulta.
  $d_d = 0$ ▷ Denominador documento.
  for doc $\in Doc_{rel}$ **do**
    $Cos_{doc} = 0$
    for indice *in* FrecuenciaTerminosConsulta **do**
      Pos_{tc} = FrecuenciaTerminosConsulta[indice][0]
      F_{tc} = FrecuenciaTerminosConsulta[indice][0]
      F_{td} = setMatrizDefrecuencia[Pos_{tc}][doc]
      if $F_{tc} \neq 0$ and $F_{td} \neq 0$ **then**
        S_{ij} = Generar Coseno de Similitud(F_{tc}, F_{td})
        S_{ii} = Generar Coseno de Similitud(F_{tc})
      else
        $S_{ij} = 1$
        $S_{ii} = 1$
        $S_{jj} = 1$
      end if
      $n+ = Sij*$ PesosConsulta[Pos_{tc}] $*$ MatrizPesos[Pos_{tc}][doc]
      $d_q+ = (S_{ii}*$ PesosConsulta[Pos_{tc}])^2
    end for
    for pD $\in$ MatrizPesos **do**
      F_{td} = obtener Matriz frecuencia(pD, doc)
      if $F_{td}! = 0$ **then**
        S_{jj} = obtener Coseno de Similitud(F_{td})
      else
        $S_{jj} = 1$
      end if
      $d_d+ = (S_{jj}*$MatrizPesos(pD, doc))^2
    end for
    $Cos_{doc}+ = \dfrac{n}{\sqrt{d_q} * \sqrt{d_d}}$ ▷ Métrica del Coseno Suave
    $L_{cos}.add(Cos_{doc})$
  end for
end procedure

4.2.1. Ejemplo del procedimiento del VSM

Para ilustrar aún mejor el procedimiento que sigue el VSM para clasificar una serie de documentos de acuerdo a la similitud de estos con una consulta, se muestra el siguiente ejemplo del modelo. En este ejemplo se tienen una serie de oraciones como representación de la colección estándar de documentos (C_{docs}) y una oración simple como representación de una consulta (q) que se aprecian en la Figura 4.5.

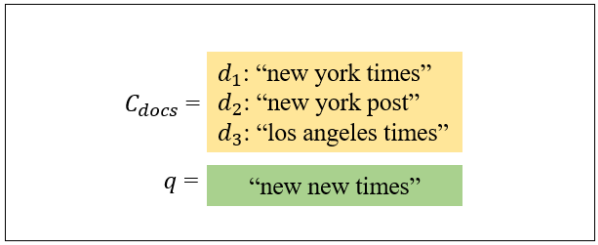

Figura 4.5: Ejemplo de una colección de documentos y una consulta para el VSM.

Una vez que se tienen la colección y la consulta se procede a generar el índice inverso (Figura 4.1) mediante la colección C_{docs}. Esta consta de extraer los términos significativos (que no sean *stopwords*) de C_{docs} y así poder indicar que documentos de la colección los contienen. Este procedimiento se puede observar en la Figura 4.6.

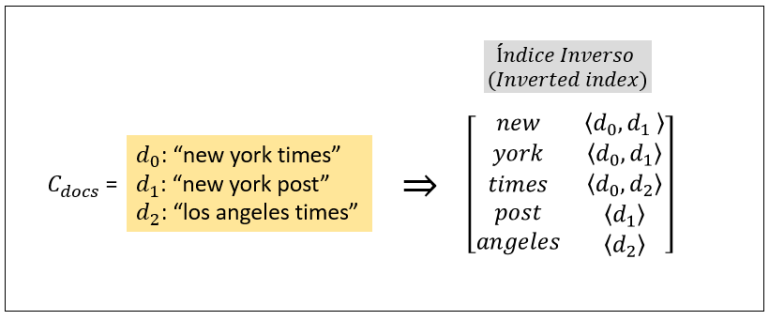

Figura 4.6: Índice inverso de la colección C_{docs}.

Este ejemplo incorpora al término *new* como vocabulario del *Invertedindex* pero en realidad este término es catalogado como *stopwords*. El índice inverso permitirá que el VSM realice la selección de documentos candidatos para una consulta de una manera más rápida y sencilla.

Como el tamaño de la colección es muy representativa los 3 documentos que componen a C_{docs} forman parte de los documentos candidatos obtenidos por el modelo mediante el *Inverted index*. Es por eso que en este ejemplo la matriz y el vector de peso y frecuencia utilizan todos los documentos de la colección C_{docs}. De lo contrario solo se utilizarían aquellos documentos que contengan al menos un término de la consulta.

Dicho lo anterior, solo faltaría proceder a obtener las matrices y vectores de pesos. Con el fin de clasificar estos documentos mediante una métrica de similitud. Como siguiente paso se procede a generar la matriz de frecuencia de la colección C_{docs} la cual se ilustra en la Figura 4.7.

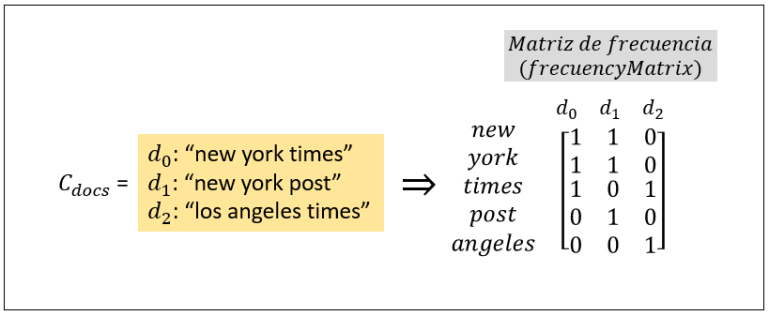

Figura 4.7: Matriz de frecuencia para la colección C_{docs}.

Aunque en la frecuencia del término de la Figura 4.7 podría interpretarse binariamente como la existencia o no de un término t_i en un documento d_j, en realidad representa las veces que el término aparece en cada documento d_j. Por ejemplo, si *york* del documento d_1 de la colección C_{docs} apareciera dos veces en este, en la matriz de frecuencia aparecería un número 2 en la posición $Frecuency\_Matrix[0][1]$.

Como las colecciones utilizadas en los modelos tienden a contener cientos de documentos, los valores en la matriz de frecuencia suelen ser muy grandes. La razón de esto podría ser debido a la excesiva frecuencia de un término t_i en un documento d_j y no por la relevancia en toda la colección D. Para esto se procede a obtener la frecuencia inversa del documento $idf_{t,d}$ (ecuación 2.10) para cada término de la colección. Los valores obtenidos al aplicar $idf_{t,d}$ se observan en la Figura 4.8.

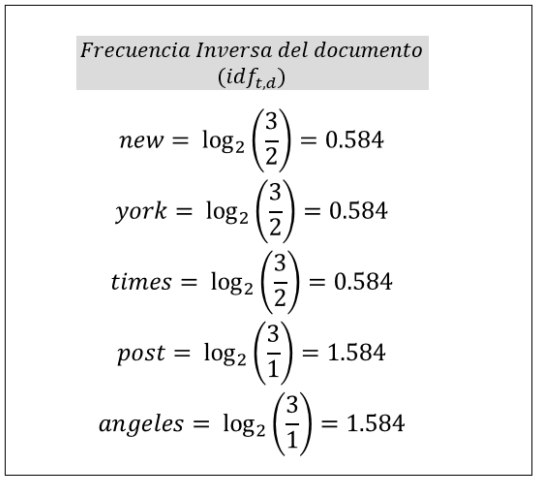

Figura 4.8: Frecuencia inversa del documento $idf_{t,d}$ de los términos significativos de la colección C_{docs}.

Los valores de la frecuencia inversa del documento y los de la matriz de frecuencia se consolidan por medio de la multiplicación para dar paso a la obtención de la matriz de pesos (ecuación 2.8). Esta ecuación permite normalizar los valores de la matriz de frecuencia, destacando a aquellos términos que son más frecuentes en toda la colección que aquellos que tienen una alta frecuencia en un documento en específico. Lo anterior se puede ilustrar en la Figura 4.9.

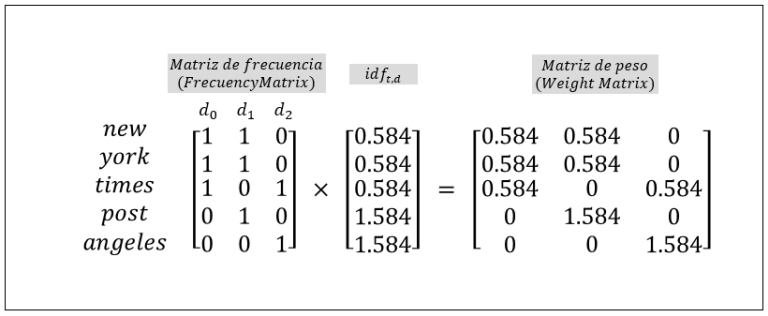

Figura 4.9: Matriz de peso $w_{t,d}$ de los términos significativos de la colección C_{docs}.

Para este punto se ha obtenido los dos factores que todo IRM debe procesar para dar pie a la clasificación de documentos: la matriz de pesos $w_{t,d}$ y el índice inverso $Inverted\_index$. Estos permiten proceder a la de búsqueda y clasificación de documentos mediante una consulta. Para este ejemplo, la consulta es la oración q ilustrada en la Figura 4.5. Para realizar la clasificación de documentos debe la consulta someterse al proceso de normalización para adquirir el vector de pesos, similar a la obtención de matriz de pesos.

El vector de pesos de la consulta está compuesta por la obtención del peso de cada término. Este peso se adquiere aplicando la ecuación 2.9. De igual forma el vector del peso debe de tener un tamaño igual a la cantidad de términos del $Inverted\_index$, debido a que las ecuaciones y estructuras están en función de la posición de estos términos, por ende todas las estructuras donde influye el vocabulario del índice inverso tienen el mismo tamaño t_i.

El primer paso para adquirir el vector de pesos es obtener el vector de frecuencia en función a los mismos términos de la consulta. La descripción y creación de este vector se puede apreciar en la Figura 4.10.

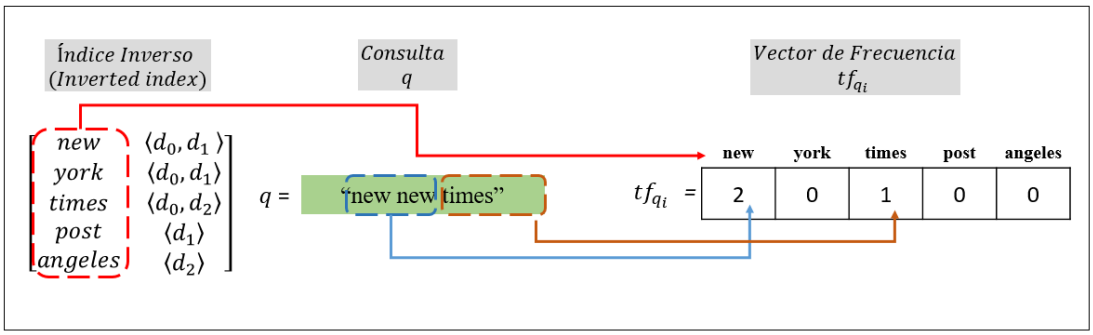

Figura 4.10: Vector de frecuencia tf_{q_i} de los términos de la consulta q.

Naturalmente en el vector de frecuencia tf_{q_i} aparecerá el valor de cero en aquellos términos que no concurran con la consulta. Obtenida el Vector de frecuencia, se procede a obtener el peso de los documentos. Para este caso no es necesario obtener la frecuencia inversa del documento, porque ya ha sido calculado en la Figura 4.8. Por lo tanto se procede a calcular el peso de cada término del vector aplicando la ecuación 2.9.

La ecuación 2.9 calcula los pesos del vector de consulta (P_{q_i}) realizando un producto entre dos factores. El primero factor es una división entre la frecuencia del término t_i y la máxima frecuencia del vector $max(tf_q)$, ambos valores obtenidos del vector de frecuencia. El segundo factor es el valor de la frecuencia inversa del documento ($idf_{t,d}$) de un término t_i. En la Figura 4.11 se puede observar los valores obtenidos de P_{q_i}.

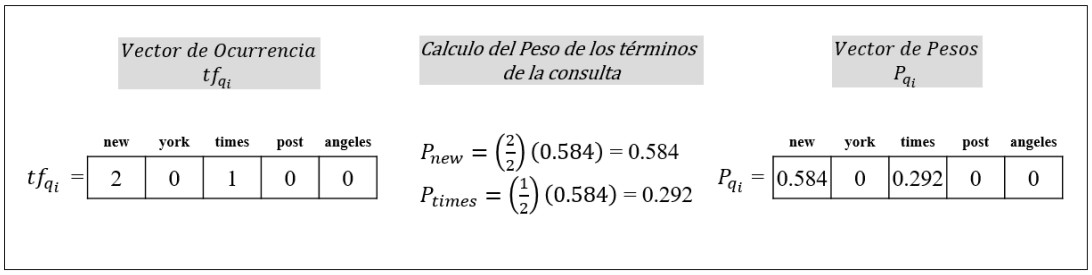

Figura 4.11: Vector de Pesos P_{q_i} de los términos de la consulta q.

Cuando se tiene el vector y la matriz de pesos se procede a la clasificación de documentos para la consulta. Para esto se utiliza una métrica de similitud. Dicha métrica es *soft cosine measure* proporcionada por [Sidorov et al., 2014]. La métrica se puede observar en la ecuación 2.7.

Esta métrica permite calcular la similitud que un documento d_j tiene con una consulta q mediante la comparativa de cada término en común de estos. La métrica considera la similitud de los términos no solo por su peso, sino también por los vectores base de cada término, los cuales serían los valores de frecuencia. Es por esto que la métrica contiene los factores S_{ii}, S_{jj} y S_{ij} para obtener la similitud de un término t_i utilizando la frecuencia de este.

EL procedimiento siguiente de clasificación muestra solo la obtención de la similitud de un documento d_j con la consulta q. Por ende el mismo procedimiento se debe realizar para cada uno de los documentos candidatos. Primeramente se muestra en la ecuación 4.1 la sustitución de los factores de *soft cosine measure* con las estructura manejadas en este ejemplo.

$$soft\_cosine_1(q, d_j) = \frac{\sum_{i,j}^{N} S_{ij} \cdot P_i \cdot W_{ij}}{\sqrt{\sum_{i}^{N} S_{ii} \cdot P_i \cdot P_i} \cdot \sqrt{\sum_{j}^{N} W_{ij} \cdot W_{ij}}} \tag{4.1}$$

$$S_{ij}, S_{jj}, S_{ii} = coseno(f_i, f_j) \tag{4.2}$$

Los factores S_{ii}, S_{ij} y S_{jj} de la ecuación 4.1 son la representación de una métrica de similitud. En este caso la métrica es la del coseno (ecuación 2.6). Esta recibe las frecuencias de los términos (f_i, f_j) (ecuación 4.2).

Sustituyendo los valores del vector y la matriz de pesos en la ecuación 4.1 de $soft\_cosine_1()$ el procedimiento para calcular el valor del numerador comparando el documento d_0 con la consulta q sería el siguiente:

$$\begin{aligned}\sum_{i,j}^{N} S_{ij} \cdot P_i \cdot W_{ij} =& coseno(tf_0, fM_{00})(P_0 \cdot W_{00}) + coseno(tf_1, fM_{10})(P_1 \cdot W_{10}) + \\ & coseno(tf_2, fM_{20})(P_2 \cdot W_{20}) + coseno(tf_3, fM_{30})(P_3 \cdot W_{30}) + \\ & coseno(tf_4, fM_{40})(P_4 \cdot W_{40}) \\ =& \cancelto{2}{coseno(2,1)}(0.584 \cdot 0.584) + \cancelto{0}{coseno(0,1)(0 \cdot 0.84)} + \\ & \cancelto{1}{coseno(1,1)}(0.292 \cdot 0.584) + \cancelto{0}{coseno(0,0)(0 \cdot 0)} + \\ & \cancelto{0}{coseno(0,0)(0 \cdot 0)} \\ =& (2)(0.3410) + (1)(00.1705) = \mathbf{0.8525}\end{aligned}$$

Como puede notar existe dentro de la métrica *soft cosine measure* una evaluación con la métrica del coseno $coseno(f_i, f_j)$. La obtención de los valores de similitud para los diferentes casos ($S_{ij}, S_{jj} y S_{ii}$) se muestran en las lineas 13, 14 y 26 del Algoritmo 9. Los métodos de esas lineas se obtienen mediante el procedimiento de los Algoritmos 7 y 8.

El siguiente paso sería el obtener los valores de los dos factores que componen al denominador de la ecuación 4.1. El primer factor a resolver es la raíz compuesto por el vector de pesos P_i. Dicho procedimiento se muestra a continuación:

$$\begin{aligned}\sqrt{\sum_{i}^{N} S_{ii} \cdot P_i \cdot P_i} =& \sqrt{coseno(tf_0, tf_0)(P_0 \cdot P_0)} + \sqrt{coseno(tf_1, tf_1)(P_1 \cdot P_1)} + \\ & \sqrt{coseno(tf_2, tf_2)(P_2 \cdot P_2)} + \sqrt{coseno(tf_3, tf_3)(P_3 \cdot P_3)} + \\ & \sqrt{coseno(tf_4, tf_4)(P_4 \cdot P_4)} \\ =& \sqrt{coseno(2,2)(0.584 \cdot 0.584)} + \cancelto{0}{\sqrt{coseno(0,0)(0 \cdot 0)}} + \\ & \sqrt{coseno(1,1)(0.292 \cdot 0.292)} + \cancelto{0}{\sqrt{coseno(0,0)(0 \cdot 0)}} + \\ & \cancelto{0}{\sqrt{coseno(0,0)(0 \cdot 0)}} \\ =& \sqrt{\cancelto{1}{coseno(2,2)}(0.584 \cdot 0.584)} + \sqrt{\cancelto{1}{coseno(1,1)}(0.292 \cdot 0.292)} \\ =& \sqrt{0.3410 + 0.0852} = \sqrt{0.2327} = \mathbf{0.4823}\end{aligned}$$

El segundo factor que compone al denominador es el que evalúa los pesos en la matriz con respecto al documento d_j. El procedimiento para obtener el valor de la segunda raíz del denominador de la ecuación 4.1 se observa a continuación:

$$\begin{aligned}
\sqrt{\sum_{i}^{N} S_{ii} \cdot W_{i0} \cdot W_{i0}} =& \sqrt{coseno(fM_{00}, fM_{00})(W_{00} \cdot W_{00})} + \sqrt{coseno(fM_10, fM_{10})(W_{10} \cdot W_{10})} + \\
& \sqrt{coseno(fM_{20}, fM_{20})(W_{20} \cdot W_{20})} + \sqrt{coseno(fM_{30}, fM_{30})(W_{30} \cdot W_{30})} + \\
& \sqrt{coseno(fM_{40}, fM_{40})(W_{40} \cdot W_{40})} \\
=& \sqrt{coseno(1,1)(0.584 \cdot 0.584)} + \sqrt{coseno(1,1)(0.584 \cdot 0.584)} + \\
& \sqrt{coseno(1,1)(0.584 \cdot 0.584)} + \sqrt{coseno(0,0)(0 \cdot 0)}^{0} + \\
& \sqrt{coseno(0,0)(0 \cdot 0)}^{0} \\
=& \sqrt{coseno(1,1)^{1}(0.584 \cdot 0.584)} + \sqrt{coseno(1,1)^{1}(0.584 \cdot 0.584)} + \\
=& \sqrt{coseno(1,1)^{1}(0.584 \cdot 0.584)} \\
=& \sqrt{(1)(0.3410) + (1)(0.3410) + (1)(0.3410)} = \sqrt{1.0231} = \mathbf{1.0114}
\end{aligned}$$

Por último se procede a terminar la evaluación sustituyendo los valores en la ecuación 4.1 y obtener la similitud del documento d_0 con la consulta q. El valor de similitud del documento d_j se muestra en la ecuación 4.3.

$$soft\_cosine_1(q, d_0) = \frac{0.8525}{0.4823 \cdot 1.0114} = \mathbf{1.7476} \tag{4.3}$$

El mismo procedimiento anterior se sigue para obtener el valor de similitud de los demás documentos d_j que forman parte de los documentos candidatos. En la siguiente secuencia de ecuaciones se muestran los valores de similitud para los documentos d_1 y d_2.

$$soft\_cosine_1(q, d_1) = \frac{0.6820}{0.6525 \cdot 1.7863} = \mathbf{0.5851} \tag{4.4}$$

$$soft\_cosine_1(q, d_2) = \frac{0.1705}{0.6525 \cdot 1.6882} = \mathbf{0.1547} \tag{4.5}$$

Los resultados de similitud obtenidos para los documentos reflejan que el documento $d_0 \in C_{docs}$ es el que obtiene la mayor similitud. Sucesivamente los documentos $d_1, d_2 \in C_{docs}$ fueron los siguientes en esta evaluación.

4.3. Ontología

El diseño e implementación de la ontología está basada en la incorporación de la familia léxica (2.2.4) de un término t_i a la consulta. Primeramente se procedió al diseño de la estructura de la ontología. Este diseño se puede observar en la Figura 4.12.

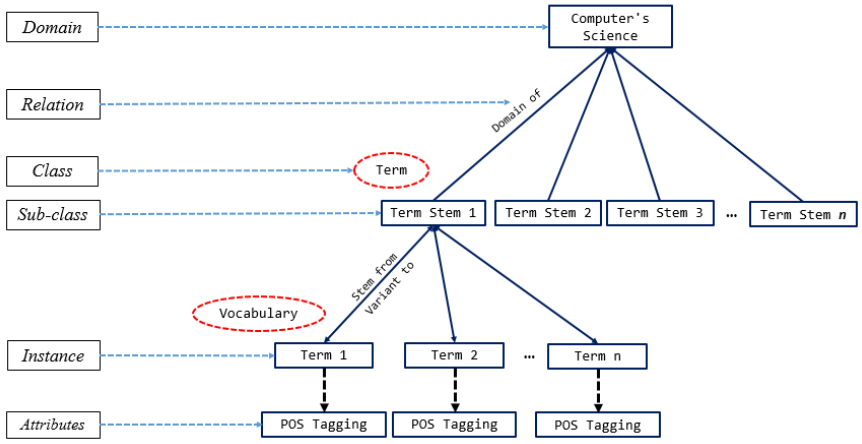

Figura 4.12: Diseño estructural de la ontología con dominio en Ciencias de la Computación.

Lo que se puede decir de la estructura de la ontología es que cada término de la clase ***Term*** será del tipo *stem*. Cada término de la clase ***Term*** tiene relacionado una familia léxica mediante las etiquetas $Variant\_to$ y $Stem\_from$. Cada término de la familia léxica es de la clase ***Vocabulary***.

Como segundo paso se procede a definir el criterio que establece la selección de los términos que componen a las familias léxicas. Tal criterio permite seleccionar aquellos términos que tienen las etiquetas gramaticales con mayor nivel de frecuencia en la colección.

Para la construcción de la ontología se requirió obtener una muestra representativa de vocabulario para construir la familia léxica de cada término. Para esto se extrajo del corpus de la herramienta *Freeling* un total de 88,837 palabras. Este corpus tiene asociada a cada palabra el tipo de palabra, el lema y su etiqueta gramatical.

El orden de prioridad en las etiquetas gramaticales se define mediante un estudio del porcentaje de frecuencia de estas en la colección. En la Tabla 4.1 se puede observar los resultados de este estudio. Donde la primera columna indica el etiquetado gramatical. La segunda columna contiene el número de veces que cada etiquetado apareció en la colección. La última columna indica el porcentaje que representa la frecuencia del etiquetado gramatical en la colección.

Tabla 4.1: Resultados del análisis de etiquetas gramaticales frecuentes en el vocabulario de la colección de documentos.

Etiquetado gramatical	***Frecuencia***	**Porcentaje**
NNP	11001	0,358047193 – 35,8 %
NN	8460	0,27534581 – 27,5 %
NNS	2635	0,085760781 – 8,6 %
CD	2258	0,073490643 – 7,3 %
JJ	1553	0,050545159 – 5,1 %
VBD	1124	0,036582587 – 3,7 %
RB	738	0,024019528 – 2,4 %
VBG	690	0,022457282 – 2,2 %
VBZ	629	0,020471928 – 2,0 %
VBP	506	0,016468674 – 1,6 %
VBN	460	0,014971522 – 1,5 %
IN	178	0,005793328 – 0,6 %
-NONE-	148	0,004816924 – 0,5 %
NNPS	73	0,002375915 – 0,2 %
VB	49	0,001594793 – 0,2 %
DT	39	0,001269325 – 0,1 %
JJR	37	0,001204231 – 0,1 %

Las etiquetas gramaticales se utilizan para elegir al conjunto de términos que forman parte de una familia léxica. Las etiquetas seleccionadas son aquellas que tiene al menos 1 % de frecuencia en la colección. El siguiente paso es seleccionar el vocabulario que compone a la ontología. El vocabulario seleccionado es extraído de la colección de documentos `CACM`.

Al vocabulario se le aplica una técnica de NLP conocida como *Pos tagging*(2.2.3). Este proceso implica obtener de cada término del vocabulario los siguientes 4 elementos:

- *stem*
- familia léxica
- *POS tagging*
- Clase

En cada término del vocabulario se extrae la palabra raíz (*stem*) para obtener su familia léxica. Esta familia se extrae de un corpus de la herramienta *Freeling*[REFERENCIA]. Finalmente el número total de términos que componen a la ontología es de 62,964.

En la Tabla 4.2 se puede observar una muestra del vocabulario de la ontología. En la primera columna se aloja el término tal cual como aparece en la colección. En la segunda columna esta la palabra raíz de ese término. En la tercera columna se encuentra la familia léxica de la palabra raíz. Por último, se encuentra la clase de palabra del término de la columna tres.

Tabla 4.2: Muestra del conjunto de palabras que se alojan en la Ontología.

Término original	***Stem***	**Familia léxica**	**Etiquetado gramatical**	**Clase**
program	program	program	NN	SUSTANTIVO
program	program	programs	NNS	SUSTANTIVO
program	program	program	VB	VERBO
program	program	programed	VBD	VERBO
program	program	programing	VBG	VERBO
program	program	programed	VBN	VERBO
program	program	program	VBP	VERBO
program	program	programs	VBZ	VERBO
processor	processor	processor	NN	SUSTANTIVO
processor	processor	processors	NNS	SUSTANTIVO

Definida la estructura, el contenido y los criterios de selección de términos se procede a implementar y obtener la ontología. El algoritmo 10 muestra el proceso general para su construcción.

Algoritmo 10 Crear ontología

```
1: procedure CREAR ONTOLOGÍA(Ontoly.owl)
2:     initialize Ontolgy.owl
3:     call to readQueryCACM(Ontology.owl)
4:     call to readOntologyModel(Ontology.owl)
5:     call to CompoundOntologyQuerysV2(Ontolgy.owl)
6: end procedure
```

El procedimiento para obtener un conjunto de términos de la ontología a partir de un término t_i de una consulta se muestra en el Algoritmo 11. Para poder extraer la familia léxica de un término t_i se envía una consulta en SPARQL a la ontología, la cual se aprecia en la línea 10.

Después se procede a extraer el resultado arrojado por la ontología para guardar la familia léxica de los términos. Este procedimiento se aprecia entre las líneas de la 11 a

la 22. El procedimiento anterior se realiza para cada termino t_i de cada consulta.

Algoritmo 11 Compound Query Ontoly Model

```
1: procedure CompoundQueryOntolyModelV2(File)
2:     finalQueries gets empty String
3:     queries ← Query_SPARQL(prefix, ontolgyModel, source, name)
4:     for q = 0 to 5 do
5:         terms[] ← from list queries to lower case split by ","
6:         originalTerms[] ← from list queries split by ","
7:         for j to terms size do
8:             finalQueries ← originalTerms[j] + ","
9:             answerQueries ← empty list
10:            answer ← Query_SPARQL(terms[j])
11:            if answer is not empty then
12:                words ← extract words from answer
13:                for ∀ k ∈ words do
14:                    pairOfWords ← words[k]
15:                    term0 = pos[j]
16:                    termS = pairOfWords[1]
17:                    if termS ≠ term0 then
18:                        answerQueries ← add pairOfWords
19:                    end if
20:                end for
21:                FinalQueries ← get first five words from answerQueries in order of
   priority
22:            end if
23:        end for
24:    end for
25:    save FinalQueries in file
26: end procedure
```

Las consultas enviadas a la ontología están previamente construidas en la Clase *Query_SPARQL* y son llamadas mediante un objeto, el cual se declara en la línea 3 del Algoritmo 11. Cada consulta recibe como parámetro el término a buscar para extraer la familia léxica.

Capítulo 5

Resultados Experimentales

5.1. Condiciones experimentales

El experimento sobre la comparativa de métricas de similitud (coseno y *soft cosine measure*) estuvo sujeta sobre un equipo de cómputo con las siguientes características físicas:

- Procesador: Intel(R) Core(TM) i5-3317U.
- RAM: 4 GB
- Almacenamiento: 417 GB
- Sistema Operativo: Windows 10, 64 bits.

Los experimentos de las tres expansiones de consulta fueron realizados sobre una estación de trabajo (*work station*) del *cluster Ehécatl* localizado en el Laboratorio Nacional de Tecnologías de Información (`LANTI`) del Instituto Tecnológico de Ciudad Madero. Esta estación de trabajo tiene las siguientes características físicas:

- Procesador: 1 Intel Xenon 8 cores.
- Memoria: 32 GB
- Almacenamiento: 2x1Tb
- Tarjeta gráfica Nvidia GTX 750, 2GB
- Sistema operativo: Windows 10

El número de consultas y documentos utilizados de la colección `CACM` para estas experimentaciones fueron de 5 y 130 documentos, respectivamente.

5.2. Comparativa de métricas de similitud

VSM utiliza usualmente la métrica del coseno para determinar los documentos relevantes de una consulta. Sin embargo, en el trabajo de [Sidorov et al., 2014] publicó que la métrica *Soft cosine Measure* arrojo mejores resultados para el problema que abordo utilizando el VSM. Por ende se procedió a realizar una evaluación para determinar cúal de estas dos métricas mostraba mejores resultados en *Precision* y *Recall*.

Esta experimentación no contempla la expansión de consultas. Lo cual indica que las 5 consultas fueron enviadas al VSM solo con la eliminación de *stopwords*, como se muestra en la Tabla 5.1. En la primera columna está el número de la consulta. En la segunda columna se aloja la consulta original. En la tercera columna se encuentra la consulta con los términos *stopwords* descartados.

Tabla 5.1: Las 5 consultas en formato original y sin términos *stopwords*

Núm	Consulta Original	sin *stopwords*
1	What articles exist which deal with TSS (Time Sharing System), an operating system for IBM computers?	**What**, articles, exist, deal, TSS, Time, Sharing, System, operating, system, IBM, computers
2	I am interested in articles written either by Prieve or Udo Pooch; Prieve, B.; Pooch, U.	I, interested, articles, written, **either**, Prieve, Udo, Pooch, Prieve, B, Pooch, U
3	Intermediate languages used in construction of multi-targeted compilers; TCOLL	Intermediate, languages, used, construction, multi, targeted, compilers, TCOLL
4	I'm interested in mechanisms for communicating between disjoint processes, possibly, but not exclusively, in a distributed environment. I would rather see descriptions of complete mechanisms, with or without implementations, as opposed to theoretical work on the abstract problem. Remote procedure calls and message-passing are examples of my interests.	I'm, interested, mechanisms, communicating, disjoint, processes, possibly, exclusively, distributed, environment, I, **would**, **rather**, see, descriptions, complete, mechanisms, **without**, implementations, opposed, theoretical, work, abstract, problem, Remote, procedure, calls, message, passing, examples, interests
5	I'd like papers on design and implementation of editing interfaces, window-managers, command interpreters, etc. The essential issues are human interface design, with views on improvements to user efficiency, effectiveness and satisfaction.	I'd, **like**, papers, design, implementation, editing, interfaces, window, managers, command, interpreters, **etc**, **The**, essential, issues, human, interface, design, views, improvements, user, efficiency, effectiveness, satisfaction

Como se puede observar en la Tabla 5.1 las palabras resaltadas en color rojo son catalogadas como *stopwords* pero debido a que el corpus de la herramienta de NLTK no incluye a estas palabras no logro detectarlas y excluirlas.

Los resultados de esta comparativa realizada para determinar cuál de las dos métricas utilizar en el IRS se puede apreciar en la Tabla 5.2. La primera columna indica el número de consulta. La segunda columna se alojan el promedio de precisión para cada consulta utilizando la métrica *soft cosine measure*. La tercera columna contiene los promedios de precisión de las consultas usando la métrica del coseno.

Tabla 5.2: Precisión promedio media (MAP) de las métricas de similitud coseno y *Soft Cosine Measure*.

Consulta	***Soft Cosine Measure***	**Coseno**
1	0.21168215	0.21168215
2	**0.7037037**	0.58496732
3	0.3999999	0.3999999
4	0.56329017	0.58152815
5	0.4618459	0.4617792
MAP	**0.397278626**	**0.39005497**

Como se puede apreciar en la Tabla 5.2 la métrica *soft cosine measure* proporciono en general un mejor valor de precisión que la métrica del coseno. La diferencia entre estas métricas es de tan solo el 0.7 %. De igual forma el *soft cosine measure* proporciono el mejor valor de precisión global, el cual se observa en la consulta dos.

5.3. Expansión de consultas

Para expandir las consultas empleando sinónimos y lexemas se utilizo el corpus de *Freeling*, y para construir la ontología con las familias léxicas se uso el corpus de *Freeling* y `CACM`. Para poder expandir una consulta primero se procede a remover las *stopwords*(2.2.2) para obtener un conjunto de términos con un aporte de información significativo.

En la Tabla 5.1 se puede observar el resultado de procesar las consultas adquiridas de la colección CACM para extraer las palabras vacías. Una vez que se obtuvieron las consultas sin *stopwords* se procedió a realizar las expansiones.

5.3.1. Familia léxica

La ontología permite incorporar un conjunto de nuevos términos a la consulta. Dicho conjunto son los términos que integran la familia léxica de cada término de las consultas. En la Tabla 5.3 se muestra un ejemplo del resultado de expandir las consultas con la ontología. La primera columna corresponde al número de consulta. La segunda a la consulta original (aún con *stopwords*). La tercera columna muestra el resultado de

la expansión usando la ontología y sin *stopwords*.

Tabla 5.3: Expansión de la consulta tres con familia léxica mediante la ontología.

Num	Consulta Original	Familia Léxica
3	Intermediate languages used in construction of multi-targeted compilers; TCOLL	Intermediate, intermediates, languages, language, used, use, using, uses, use, construction, constructions, multi, targeted, target, targets, targetted, targetting, targeting, compilers, compiler, TCOLL

Los términos resaltados en color azul son aquellos que fueron extraídos de la ontología e incorporados como familia léxica. En algunos casos se logró expandir hasta 5 términos, siendo que por definición del objetivo el límite es de 5.

5.3.2. Sinónimos

La expansión con sinónimos, se realizó incorporando un máximo de 5 sinónimos por cada término de la consulta. Para determinar el grupo de sinónimos de un término se colocó como cláusula en la búsqueda que obtuviera los sinónimos del término t_i que compartieran el mismo lexema, buscando con ello una relación semántica afín.

Debido a que la búsqueda de sinónimos en el corpus *Freeling* a porto muchos sinónimos, se decidió acotar dicho conjunto a un máximo de 15 y a partir de este conjunto elegir aleatoriamente un máximo de 5. Un ejemplo de esta expansión se muestra en la Tabla 5.4. La primera columna corresponde al número de la consulta. La segunda muestra la consulta original (aún con *stopwords*) y la tercera el resultado de la expansión con sinónimos.

Tabla 5.4: Expansión de la consulta número tres con términos de la clase *sinónimos*.

Num	Consulta Original	Sinónimos
3	Intermediate languages used in construction of multi-targeted compilers; TCOLL	intermediate, middle, chemical, sophomore, next-to-last, third-year, languages, speech, communication, nomenclature, terminology, text, used, usage, utility, usefulness, purpose, custom, construction, building, interpretation, cerebration, intellection, artifact, multi, targeted, place, mark, victim, butt, end, compilers, encyclopedist, encyclopaedist, programme, writer, author, TCOLL

En este caso se puede observar que en casi todos los términos de la consulta se logró incorporar los 5 términos correspondientes a la expansión de sinónimos.

5.3.3. Lexema o raíz

La expansión por lexema está referida a incorporar solo el *stem* de cada término de las consultas. Para este procedimiento se utilizó el corpus de *Freeling*. Un ejemplo de esta expansión se muestra en la Tabla 5.5. La primera columna corresponde al número de consulta. La segunda a la consulta original (aún con *stopwords*) y la tercera columna muestra el resultado de la expansión con lexema.

Tabla 5.5: Expansión de la consulta número tres con lexemas.

Num	Consulta Original	Lexema
3	Intermediate languages used in construction of multi-targeted compilers; TCOLL	Intermediate, languages, language, used, use, construction, multi, targeted, target, compilers, compiler, TCOLL

Los términos resaltados en color azul son aquellos que fueron extraídos de *Freeling*. En este caso solo se incorpora la raíz del término. Los términos de la columna 2 de la Tabla 5.5 fueron expandidos siempre y cuando no fuese en sí un término raíz.

5.4. Resultados obtenidos

Aplicando las expansiones descritas en la sección 5.3 a cada una de las 5 consultas que formaron parte de la experimentación de este trabajo se obtuvieron los siguientes resultados en *Recall* y *Precision*.

En la Tabla 5.6 se puede observar que para los tres diferentes métodos de expansión y las consultas originales se recuperaron todos los documentos relevantes por el IRS. Es por eso que los resultados son del 100 %.

Tabla 5.6: *Recall*: Porcentaje de documentos relevantes recuperados por IRS.

Num	Original	Lexema	Familia léxica (Ontologia)	Sinónimos
1	100 %	100 %	100 %	100 %
2	100 %	100 %	100 %	100 %
3	100 %	100 %	100 %	100 %
4	100 %	100 %	100 %	100 %
5	100 %	100 %	100 %	100 %

Para adquirir la precisión promedio de una consulta se toma en cuenta la posición en donde esta el documento relevante. Los documentos relevantes aparecerán en el

conjunto de documentos recuperados por el IRS . El procedimiento de la precisión promedio se puede apreciar en la sección 2.3.4.

En la Tabla 5.7 se muestra en la primera columna el número de la consulta. La segunda columna muestra la precisión que obtuvo el modelo sin ningún tipo de expansión. La tercera columna muestra la precisión de la expansión con el lexema. La cuarta columna muestra los resultados de precisión al expandir la consulta por medio de la familia léxica usando una ontología. Finalmente en la quinta columna la precisión de la expansión con sinónimos. Y en la última fila se observa la media del promedio de precisión (MAP) de cada consulta en las diferentes expansiones.

Tabla 5.7: Precisión promedio y MAP de las consultas.

Num	Original	Lexema	Familia léxica (Ontologia)	Sinónimos
1	0.468695652	0.415714286	**0.481597796**	0.293859649
2	**0.519230769**	0.473015873	0.108469872	0.175418275
3	0.192411178	0.131414558	0.12816961	**0.218704963**
4	**0.517710547**	0.424539948	0.323046001	0.300445916
5	**0.488045635**	0.458372442	0.418659547	0.340469711
MAP	**0.437218756**	0.380611421	0.291988565	0.265779703

Los valores resaltados en color rojo de cada fila indican cuál de las diferentes expansiones arrojaron la mejor precisión para las diferentes consultas. La expansión que utiliza la ontología muestra la mejor precisión para la consulta número uno. Así mismo la expansión mediante los sinónimos muestra la mejor precisión para la consulta número tres.

También se utilizo la métrica de proposición de fallo (*Fall-out*) la cual mide que porcentaje de documentos no relevantes están incluidos en los documentos recuperados.

En la Tabla 5.8 se observan el *Fall-out* de cada consulta para las diferentes expansiones. En la primera columna está el número de la consulta. En la segunda columna se muestra el porcentaje de fallo de la consulta original. En la tercera columna se muestra el porcentaje de fallo con la expansión con lexema. En la cuarta columna se muestra el porcentaje de fallo con respecto a la expansión de la familia léxica usando la ontología. Por ultimo en la columna 5 está el porcentaje de fallo de la expansión con sinónimos.

Tabla 5.8: Proposición de fallo (*Fall-out*)

Num	Original	Lexema	Familia léxica (Ontología)	Sinónimos
1	90 %	95 %	95 %	98 %
2	99 %	98 %	99 %	98 %
3	**89 %**	97 %	98 %	98 %
4	**100 %**	**100 %**	**100 %**	99 %
5	98 %	96 %	98 %	99 %

Para poder determinar si existe una diferencia significativa en las diferentes expansiones con las consultas originales se prosigue a realizar la prueba de Wilcoxon. Esta prueba permite conocer si existe una diferencia estadística significativa entre un conjunto de datos A y un conjunto de datos B.

Los resultados de esta prueba se pueden observar en la Tabla 5.9. La primera columna es el conjunto A a comparar. La segunda columna es el conjunto B a ser comparado. La tercera columna muestra el valor de *p-value*.

Tabla 5.9: Prueba de Wilcoxon para determinar la diferencia estadistica en las diferentes expansiones.

A	**B**	***p-value***
Original	Lexema	0.1320
Original	Familia léxica (Ontología)	0.06494
Original	Sinónimos	**0.04113**
Lexema	Familia léxica (Ontología)	0.3095
Lexema	Sinónimos	0.06494
Familia léxica (Ontología)	Sinónimos	0.8182

En la Tabla 5.9 muestra la comparación entre los tres tipos de métodos de expansión utilizados, dado estos resultados se puede decir que solamente existe una diferencia estadística significativa entre la consulta original y la consulta expandida mediante sinónimos. Todas las otras comparaciones son estadisticamente no significativas.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se compararon tres diferentes métodos de expansión:(a) Familia léxica, (b) sinónimos y (c) lexema, utilizando el dominio en Ciencias de la Computación. El IRM seleccionado para realizar la búsqueda y recuperación de documentos en este trabajo fue el Modelo de Espacio Vectorial (VSM) utilizando *Soft Cosine Measure* como métrica de clasificación, el cual empleo una muestra de la colección `CACM`. Esta muestra está compuesta por 5 consultas y 130 documentos.

El IRS al final de la fase experimental se obtuvo que los tres diferentes métodos de expansión obtuvieron un *Recall* de 100 %, esto significa que se pudo extraer el 100 % de los documentos relevantes para cada una de las 5 consultas.

Para explicar los resultados de la métrica de precisión se realizo una prueba de Wilcoxon que compara los diferentes métodos de expansión y la consulta original. Dicha prueba obtiene que los resultados son estadisticamente no significativos a excepción del resultado obtenido entre la consulta original y el método de expansión por sinónimos.

6.2. Principales aportaciones

- Implementación del Modelo de Espacio Vectorial, el cual puede ser utilizado en otros trabajos que sigan esta línea de investigación.
- Comparación sobre la precisión de las métricas de similitud del coseno y *soft cosine measure* utilizando las consultas y documentos empleadas en este trabajo de tesis.
- Construcción de una ontología con dominio en Ciencias de la Computación de la colección CACM.

6.3. Publicaciones y ponencias

- Exposición de ponencia en el VIII Encuentro de Investigadores en el Instituto Tecnológico de Ciudad Madero Campus 2, del 1 al 4 de Diciembre con el tema "Implementación de un prototipo de un sistema de recuperación de información que utilice ontologías para la expansión de consultas"
- Envió de un artículo con el tema *Implementation of an Information Retrieval System Using the Soft Cosine Measure* para el ISCI (International Seminar on Computational Intelligence) del Instituto Tecnológico de Tijuana. Participantes: González Barbosa, Frausto Solís, Terán Villanueva, Castilla Valdés, Florencia Juárez, Hernández González , Mojica Mata. (Enero de 2016).

6.4. Trabajo futuro

1. Aumentar número de consultas, por lo tanto se incrementa la colección.
2. Diseñar la ontología con diferente estructura. Por ejemplo este trabajo utilizó familias léxicas, mientras que en el trabajo de [Kuna et al., 2014] utiliza taxonomía del dominio y el trabajo de [Valbuena and Londoño, 2014] crea un árbol global de términos mediante el uso de diferentes ontologías y la colección de documentos.
3. Cambiar los criterios de selección de sinónimos. Por ejemplo, incorporar los primeros 5 a la consulta.

Bibliografía

[Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). Modern information retrieval, volume 463. ACM press New York.

[Benjamins et al., 1998] Benjamins, R., Fensel, D., and Gómez-Pérez, A. (1998). Knowledge management through ontologies. CEUR Workshop Proceedings (CEUR-WS. org).

[Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). Natural Languaje Processing with Python. O'Reilly Media, Inc.

[Carbonell, 2003] Carbonell, J. (2003). El procesamiento del lenguaje natural, tecnología en transición.

[Ding et al., 2006] Ding, C., Li, T., and Peng, W. (2006). Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In Proceedings of the national conference on artificial intelligence, volume 21, page 342. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[E. Loos et al., 2003] E. Loos, E., Anderson, S., Day. Jr, D. H., C. Jordan, P., and Douglas Wingate, J. (2003). Glossary of linguistic terms.

[Farrús and Costa-jussà, 2013] Farrús, M. F. and Costa-jussà, M. R. (2013). Evaluación automática del aprendizaje electrónico utilizando el análisis semántico latente: un caso de uso. Revista Mexicana de Bachillerato a Distancia, 5(10):153–165.

[G. Jorán, 2003] G. Jorán, A. (2003). Lenguas y tecnologías de la información.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2):199–220.

[Guarino, 1999] Guarino, N. (1999). The role of identity conditions in ontology design. In International Conference on Spatial Information Theory, pages 221–234. Springer.

[Gudivada et al., 1997] Gudivada, V. N., Raghavan, V. V., Grosky, W. I., and Kasanagottu, R. (1997). Information retrieval on the world wide web. IEEE Internet Computing, 1(5):58.

[Heflin et al., 1997] Heflin, J., Hendler, J., Luke, S., Gasarch, C., Zhendong, Q., Spector, L., and Rager, D. (1997). Simple HTML Ontology Extensions.

[Kuna et al., 2014] Kuna, H., Martin, R., Martini, E., and Solonezen, L. (2014). Desarrollo de un Sistema de Recuperació n de Informació n para Publicaciones

Cient{\'\i}ficas del Á rea de Ciencias de la Computació n. Revista Latinoamericana de Ingenieria de Software, 2(2):107–114.

[la Serna Palomino et al., 2013] la Serna Palomino, N., Pró Concepción, L., and Román Concha, U. (2013). Diseño de un sistema de recuperación de imágenes de individuos malhechores para seguridad ciudadana. Revista de investigación de Sistemas e Informática, 10(1):25–32.

[La Serna Palomino et al., 2009] La Serna Palomino, N., Román Concha, U., and Osorio Beltrán, N. (2009). Implementación de un Sistema de Recuperación de Información. Revista de investigación de Sistemas e Informática, 6(1):57–64.

[Lobo, 2010] Lobo, C. (2010). Educación Adultos: Ámbito Comunicación II. Lengua castellana y Literatura:.

[Manning et al., 2008] Manning, C. D., Raghavan, P., Schütze, H., and Others (2008). Introduction to information retrieval, volume 1. Cambridge university press Cambridge.

[Martínez Méndez, 2004] Martínez Méndez, F. J. (2004). Recuperación de información: modelos, sistemas y evaluación. Murcia: Kiosko, 2004.

[Maurice de Kunder, 2015] Maurice de Kunder (2015). WWW size.

[Miniwatts Marketing Group., 2015] Miniwatts Marketing Group. (2015). No Title.

[Monsalve, 2012] Monsalve, L. S. G. (2012). Experimento de recuperación de información usando las medidas de similitud coseno, jaccard y dice. TECCIENCIA, 6(12):14–24.

[Montoya Pérez, 2012] Montoya Pérez, J. (2012). Construcción de un árbol de términos latentes y su uso en el cálculo de la semejanza de documentos. PhD thesis, Instituto Politécnico Nacional. Centro de Investigación en Computación.

[Neches et al., 1991] Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling technology for knowledge sharing. AI magazine, 12(3):36.

[Pabón et al., 2014] Pabón, O. S., ELENA, M., and GONZÁLEZ, S. M. (2014). PROPUESTA PARA EXTENDER SEMÁ NTICAMENTE EL PROCESO DE RECUPERACIÓ N DE INFORMACIÓ N. Revista EIA, 11(22):51–65.

[Paice, 1994] Paice, C. D. (1994). An evaluation method for stemming algorithms. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 42–50. Springer-Verlag New York, Inc.

[Sedeño, 2011] Sedeño, R. O. L. (2011). Clasificación y agrupamiento de textos de noticias. Serie Cient{\'\i}fica, 4(3).

[Sidorov et al., 2014] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., and Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas, 18(3):491–504.

[Singhal, 2001] Singhal, A. (2001). Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4):35–43.

[UNAM, 2015] UNAM (2015). Taller de lectura, redaccion e iniciacion a la investigacion documental 2.

[Valbuena and Londoño, 2014] Valbuena, S. J. and Londoño, J. M. (2014). Búsqueda de documentos basada en el uso de índices ontológicos creados con mapreduce/document search supported on an ontological indexing system created with mapreduce. Ciencia e Ingeniería Neogranadina, 24(2):57.

[Van Heijst et al., 1997] Van Heijst, G., Schreiber, A. T., and Wielinga, B. J. (1997). Using explicit ontologies in KBS development. International journal of human-computer studies, 46(2):183–292.

[W3, 2014] W3 (2014). Resource Description Framework.

[W3C, 2004] W3C (2004). OWL Web Ontology Language Guide.

[Wilbur and Sirotkin, 1992] Wilbur, W. J. and Sirotkin, K. (1992). The automatic identification of stop words. Journal of information science, 18(1):45–55.