PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO DE UN MODELO DE RECUPERACIÓN DE INFORMACIÓN USANDO EXPANSIÓN DE CONSULTAS BASADAS EN ONTOLOGÍAS EN EL DOMINIO DE LA CIENCIA DE LA COMPUTACIÓN

Tesis para optar el Título de **Ingeniero Informático,** que presenta el bachiller:

Bonnie Gabriela Carranza Chávez

ASESOR: Héctor Andrés Melgar Sasieta

Lima, noviembre de 2014

RESUMEN

A lo largo de los años, y particularmente a partir de la aparición del Internet, se ha venido dando un aumento en la cantidad de información disponible para ser consultada por las personas.

Sin embargo, la aparición de los sistemas de recuperación de información ha contribuido a facilitar la búsqueda de información para los usuarios, disminuyendo los tiempos invertidos en dicha búsqueda, y hasta cierto punto, mejorando la relevancia de la información recuperada. Sin embargo, se ha identificado que aún persisten algunos elementos que dificultan la obtención de resultados relevantes tales como características propias del lenguaje natural como ambigüedad, desconocimiento del usuario respecto a qué puede ser relevante para él, entre otros.

Ante esto, en el presente proyecto se propone una alternativa de solución de forma tal que los documentos recuperados sean en mayor medida relevantes. Esta recuperación se tratará bajo el enfoque específico de la expansión de consultas, proceso para el cual se emplearán modelos de conocimiento como lo son las ontologías.

FACULTAD DE
CIENCIAS E INGENIERÍA
ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA

PONTIFICIA
UNIVERSIDAD CATÓLICA
DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO: Diseño de un modelo de recuperación de información usando expansión de consultas basadas en ontologías en el dominio de la ciencia de la computación

ÁREA: Ciencias de la computación

PROPONENTE: Héctor Andrés Melgar Sasieta

ASESOR: Héctor Andrés Melgar Sasieta

ALUMNO: Bonnie Carranza Chávez

CÓDIGO: 20084316

TEMA N°: S 36

FECHA: 08 de setiembre de 2014

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Facultad de Ciencias e Ingeniería
07 OCT. 2014
MIGUEL MEJIA PUENTE
DECANO

DESCRIPCIÓN

El presente proyecto brinda una alternativa de solución al problema de recuperación de información con la mínima asistencia del usuario, de forma tal que los documentos recuperados sean relevantes. Esta recuperación se tratará bajo el enfoque específico de la expansión de consultas, proceso para el cual se emplean modelos de conocimiento como lo son las ontologías.

OBJETIVO GENERAL

Diseñar un modelo de recuperación de información usando expansión de consultas basadas en ontologías en el dominio de una currícula universitaria en el área de ciencias de la computación.

OBJETIVOS ESPECÍFICOS

- 1. Diseñar un mecanismo que permita derivar la consulta del usuario en una consulta más específica en un dominio.
- 2. Diseñar un mecanismo que permita resolver la ambigüedad de las palabras en un dominio en específico.
- 3. Diseñar un mecanismo que permita al usuario estructurar consultas sin la necesidad de comprender el dominio completo del conocimiento y el dominio del motor de búsqueda.
- 4. Diseñar un mecanismo que permita al sistema IR comprender las necesidades de información del usuario.
- 5. Diseñar un mecanismo que permita la recuperación por contenido semántico y no por contenido sintáctico.

Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono:
(511) 626 2000 Anexo 4801

ALCANCE

El proyecto pertenece al área de ciencias de la computación, y en particular al sub-área de ingeniería del conocimiento.

Para este proyecto se eligió diseñar un modelo en vez de crear un software (motor de recuperación), ya que el diseño del modelo permite enfocarse más en la definición de los mecanismos de recuperación y de la interacción de los componentes que en el proceso de desarrollo del software en sí. Asimismo, el diseño del modelo permite que el mismo pueda ser implementado utilizando herramientas, independientemente de la plataforma tecnológica.

Por otro lado, con la finalidad de probar el modelo se utilizan herramientas ya existentes que faciliten la configuración de un motor de recuperación que opere bajo el modelo lógico propuesto.

Para este proyecto, se ha excluido del alcance los procedimientos específicos relacionados a la selección y extracción automática de información de fuentes online, y en su lugar se utilizará información específica ingresada manualmente a la herramienta por ser considerado suficiente para realizar las pruebas de los entregables acorde al modelo diseñado.

De forma análoga, el alcance excluye la creación de un mecanismo de captura de información del usuario, por considerarse un tema amplio que podría desarrollarse en trabajos futuros. Por esa razón, para las pruebas de los entregables se empleará información (conocimiento) codificada de un usuario manualmente ingresada (información del usuario no extraída automáticamente).

Finalmente, se ha excluido del alcance el análisis y pre-procesamiento de la consulta inicial del usuario por lógica proposicional, por considerarse un tema extenso que va más allá de la problemática inicial planteada para este proyecto.

Máximo: 100 páginas

Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono
(511) 626 2000 Anexo 4801

DEDICATORIA

Quiero dedicar este proyecto de fin de carrera a mi familia, a mis padres Jorge y Norma que a lo largo de toda una vida llena de ejemplos de esfuerzo y perseverancia me han impulsado a avanzar con la alegría de quien disfruta lo que realiza.

A mi hermanos, a Normita por todo el apoyo brindado y por el gran ejemplo de empeño y profesionalismo que sigue siendo para mí, y a Carlitos por la infinita alegría que irradia siempre su compañía.

AGRADECIMIENTOS

Quiero agradecer muy especialmente al Dr. Andrés Melgar, por su gran apoyo y asesoría en todas las etapas de este proyecto. Al Dr. Manuel Tupia por el apoyo y las enseñanzas brindadas a lo largo de mi carrera. A mis amigos Héctor, Rodrigo y Martin, con quienes colaboramos en el transcurso del desarrollo de nuestros proyectos, y a Javier, Daiana, Hernan y Rodolfo que me han acompañado en todo momento.

INDICE

CAPÍTULO 1

1 Problemática

El crecimiento continuo de la información es un hecho que se ha ido consolidando a lo largo de los años debido a la necesidad de las personas de la misma. Si bien este crecimiento se ha dado desde incluso antes de la aparición de las computadoras y el inicio de la era de la tecnología, la aparición del Internet dio pie a una explosión en este crecimiento debido a la facilidad que brindaba a los usuarios de poder compartir y tener acceso, a través de este medio, a la información que necesitara sin las limitaciones espaciales tradicionales [WANG, et al 2012; CERULO, CANFORA 2004; [MITRA, SINGHAL et al 2000; PAHA, GULATI et al 2009].

Desde el punto de vista de un usuario, la forma tradicional de satisfacer esta necesidad de información– y la más precisa – podría ser consultar directamente el contenido de libros y documentos albergados tanto en bibliotecas y repositorios físicos como digitales, proceso en el cual el usuario iría reteniendo sólo aquellos documentos que considerara relevantes y que pudieran potencialmente servir para atender su necesidad identificada de información. Sin embargo, es claro que este proceso conllevaría una inversión extremadamente grande no sólo de tiempo, sino de esfuerzo, por lo cual esta alternativa de revisión exhaustiva de la documentación por parte del usuario se hace difícilmente factible, si no es físicamente imposible [RIJSBERGEN, 1979].

Gracias a que el Internet se ha ido consolidado hasta la actualidad como el repositorio más grande de información (alrededor de un trillón de páginas), en este proceso de consolidación han ido apareciendo motores de búsqueda que han logrado aliviar, hasta cierto punto, la dificultad de las personas de encontrar entre todo ese volumen de información, aquella que necesitan en determinado momento. Sin embargo, si bien estos mecanismos tecnológicos de búsqueda han reducido en gran medida el difícil proceso que representaba la alternativa de revisión exhaustiva tradicional, las necesidades y requerimientos de las personas respecto a la búsqueda también han ido evolucionando hacia expectativas más altas [KEVIN, 2010; WANG, et al 2009; GREENGRASS, 2009; ALI, KHAN 2008].

Los usuarios de los sistemas de búsqueda y recuperación de información ahora ya no sólo esperan que el sistema reduzca la cantidad de documentos sobre los cuales ellos puedan buscar la información que necesitan, sino que esperan que brindándoles una

consulta (*query*) los sistemas sean capaces, de alguna forma, de interpretar y "entender" lo que dicho usuario necesita, de tal manera que puedan avanzar un paso más brindándoles documentos con información relevante para ellos. Sin embargo, los sistemas de recuperación de información no necesariamente presentan este comportamiento ideal. Una de las razones es que el sistema no siempre logra interpretar efectivamente lo que el usuario quiere y necesita realmente [ALI, KHAN 2008; WANG, et al 2012].

En la mayoría de los modelos y sistemas actuales de recuperación de información, sólo cuando las palabras expresadas en la consulta del usuario aparecen en el documento, este documento se entrega como resultado a la búsqueda. Sin embargo, en la realidad usualmente un mismo concepto tiene varias formas de expresarse en lenguaje natural.
Como resultado de esto, es muy común que documentos relacionados y, en realidad, relevantes para el usuario sean omitidos, debido a la diferencia entre cómo se encuentra expresado en el documento y cómo fue planteado por el usuario. Este es uno de los principales problemas que enfrentan los sistemas actuales para que puedan calificarse como realmente efectivos [WANG, et al 2009; RIJSBERGEN, 1979].

Como se mencionó anteriormente, son los usuarios los que informan su necesidad de información a través de una expresión de consulta y lo hacen según como ellos logran estructurar sus ideas en frases usualmente redactadas en lenguaje natural. Esto da cabida a un segundo problema para los sistemas de recuperación, ya que se ven en la necesidad de tener que interpretar palabras escritas en este lenguaje y realizar la búsqueda sobre los documentos que conforman las fuentes de conocimiento, en su mayoría también redactados en lenguaje natural, lo cual implica lidiar con las características propias de éste, tales como: ambigüedad en las palabras, dependencia del contexto, el hecho de que una misma palabra puede tener varios significados propios de áreas o dominios de conocimiento diferentes, el hecho de que un mismo concepto puede ser expresado por distintas palabras, entre otros [BAEZA-YATES, RIBERIRO-NETO 1999; BAI, et al 2007; JALALI, BORUJERDI 2008].

Dicho en palabras de Rijsbergen, *"la necesidad del software de intentar replicar el proceso humano de 'leer' no es un problema fácil ya que la lectura involucra extraer información tanto sintáctica como semánticamente, y no sólo eso, sino también saber cómo usarlo para decidir la relevancia de lo que se es leído"* [RIJSBERGEN, 1979]. Intentos fallidos en este proceso de interpretación del lenguaje natural por parte del

sistema puede generar que no sólo se omita información que pudiera realmente sí ser relevante para el usuario, sino que también podría ser que se le entregue al usuario información excesiva que no cumpla con satisfacer sus requerimientos [WANG, et al 2009; SONG, et al 2005].

A esto se le puede sumar una tercera dificultad que ya no se basa en una característica propia del lenguaje natural, sino un poco más enfocada al usuario y al sistema.

Esto se da ya que es posible que el usuario no siempre logre traducir sus palabras en una consulta 'adecuada' por dos motivos.

En primer lugar, ya que podría deberse a dificultades propias del usuario al momento de estructurar sus ideas o el mismo hecho de que no cuentan con el conocimiento completo del dominio, por lo que no pueden especificar las palabras clave adecuadas para una consulta válida. En segundo lugar, ya que una consulta que podría definirse como adecuada para un sistema puede no serlo para otro, debido a las distintas estrategias empleadas por los sistemas al intentar realizar la interpretación de la consulta del usuario [WANG, et al 2009].

Los sistemas de *Information Retrieval* (IR) han intentado minimizar el problema de poder "comprender" la consulta, las fuentes y el contexto de la búsqueda para una recuperación efectiva de distintas formas.

Un enfoque es el de personalización, el cual consiste en que el sistema almacene el historial de consultas y documentos visualizados por el usuario y reúse esta información en futuras búsquedas.

Otro enfoque es a través de "modelos de lenguaje" que se basa en modelos estadísticos de lenguaje, de tal forma que un modelo de lenguaje es una distribución probabilística que captura las regularidades del lenguaje natural, como por ejemplo la probabilidad de ocurrencia de una palabra junto a otra.

Un tercer enfoque es el de expansión de consulta, enfoque a través del cual se busca añadir nuevos términos significativos a la consulta inicial (ya sea manual, automática o asistida por el usuario) de tal forma que se puedan obtener mejores resultados en la recuperación [BHOGAL, MACFARLANE et al 2007; JALALI, BORUJERDI 2008].

El presente proyecto plantea brindar una alternativa de solución al problema de recuperación de información con la mínima asistencia del usuario, de forma tal que los documentos recuperados sean relevantes. Esta recuperación se tratará bajo el

enfoque específico de la expansión de consultas, proceso para el cual se emplearán modelos de conocimiento como lo son las ontologías.

1.1 Árbol de problemas

A partir de la problemática anteriormente expuesta, se ha estructurado a modo de resumen en el siguiente árbol de problemas:

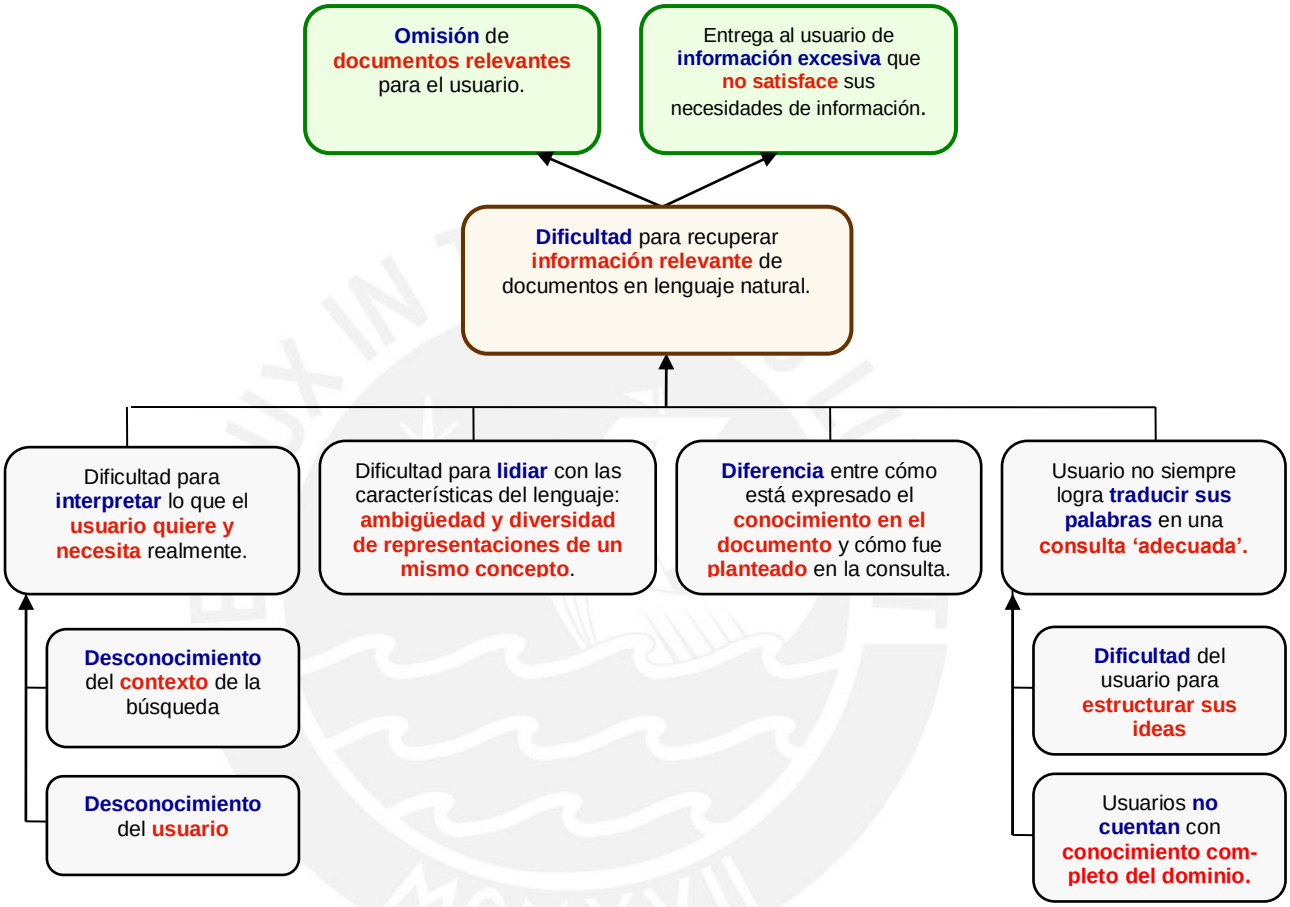


Imagen 1. Árbol de problemas

1.2 Objetivo general

Diseñar un modelo de recuperación de información usando expansión de consultas basadas en ontologías en el dominio de la ciencia de la computación.

1.3 Objetivos específicos

1. Diseñar un mecanismo que permita derivar la consulta del usuario en una consulta más específica en un dominio.
2. Diseñar un mecanismo que permita resolver la ambigüedad de las palabras en un dominio en específico.

3. Diseñar un mecanismo que permita al usuario estructurar consultas sin la necesidad de comprender el dominio completo del conocimiento y el dominio del motor de búsqueda.
4. Diseñar un mecanismo que permita al sistema IR comprender las necesidades de información del usuario.
5. Diseñar un mecanismo que permita la recuperación por contenido semántico y no por contenido sintáctico.

1.4 Resultados esperados

- Resultado 1 para el objetivo 1: Mecanismo de pre-procesamiento de textos en lenguaje natural que remueva las palabras vacías como artículos y conectores (*stopwords*).
- Resultado 2 para el objetivo 1: Mecanismo de pre-procesamiento de la consulta en lenguaje natural que reduzca las palabras a su forma base o lema (proceso de lematización).
- Resultado 3 para el objetivo 2: Modelo de desambiguación de palabras en lenguaje natural dentro de un dominio usando ontologías.
- Resultado 4 para el objetivo 3: Estructura para almacenar el conocimiento del dominio.
- Resultado 5 para el objetivo 4: Estructura para almacenar el conocimiento previo del usuario.
- Resultado 6 para el objetivo 4: Modelo que permita recuperar el conocimiento previo del usuario a partir de una estructura en la que se encuentra codificada.
- Resultado 7 para el objetivo 5: Mecanismo de expansión empleando equivalencias de conceptos.
- Resultado 8 para el objetivo 5: Modelo de interpretación y conversión de la consulta del usuario usando ontologías para la interpretación y expansión de consultas para la conversión.
- Resultado 9 para el objetivo 5: Modelo que permita la recuperación de información relevante dada una consulta.

2 Herramientas, métodos, metodologías y procedimientos

2.1 Introducción

En este apartado se presentarán las herramientas, métodos y procedimientos que serán utilizados para el desarrollo de los resultados esperados así como una breve descripción de cada uno. El lenguaje de programación que se empleará de modo transversal para el desarrollo de los mecanismos de prueba de los entregables será Java, y el entorno de desarrollo será Eclipse. Asimismo, se tiene la ventaja de que todas las herramientas tecnológicas utilizadas son gratuitas y de código abierto.

Resultados esperado	Herramientas a usarse
RE1: Mecanismo de pre-procesamiento de textos en lenguaje natural que remueva las palabras vacías como artículos y conectores (*stopwords*).	Lucene
RE2: Mecanismo de pre-procesamiento de la consulta en lenguaje natural que reduzca las palabras a su forma base o lema (proceso de lematización).	Freeling
RE3: Modelo de desambiguación de palabras en lenguaje natural dentro de un dominio usando ontologías.	OWL: Web Ontology Language SPARQL Query Languaje Jena Ontology API
RE4: Estructura para almacenar el conocimiento del dominio.	OWL: Web Ontology Language
RE5: Estructura para almacenar el conocimiento previo del usuario.	OWL: Web Ontology Language
RE6: Modelo que permita recuperar el conocimiento previo del usuario a partir de una estructura en la que se encuentra codificada.	OWL: Web Ontology Language
RE7: Mecanismo de expansión empleando equivalencias de conceptos.	SPARQL Query Languaje Jena Ontology API
RE8: Modelo de interpretación y conversión de la consulta del usuario usando ontologías para la interpretación y expansión de consultas para la conversión.	Lucene
RE9: Modelo que permita la recuperación de información relevante dada una consulta.	Lucene

Tabla 1. Mapeo de herramientas por resultado esperado.

2.2 Herramientas

Todas las herramientas serán utilizadas en sus versiones compatibles con el lenguaje Java.

- OWL

  OWL (*Web Ontology Language*) es un lenguaje diseñado para ser usado por aplicaciones que necesitan procesar el contenido de información. OWL puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos, representación también conocida como ontología. OWL tiene mayor capacidad para expresar significados y semántica que XML, RDF y RDF-S, de este modo OWL va más allá de estos lenguajes respecto a su capacidad para representar contenido interpretable por un computador en la Web [W3C, 2009].

- JENA

  Apache Jena es un framework de Java gratuito y de código abierto para la construcción de aplicaciones web semánticas y de datos enlazados. Este framework está compuesto por diferentes APIs que interactúan para procesar información contenida en ontologías.

  Una de las APIs que brinda este framework es la Jena Ontology API, la cual provee una interfaz de programación que facilita la carga de ontologías en diversos lenguajes, entre ellos OWL, para su posterior tratamiento.

- SPARQL

  El lenguaje de consultas SPARQL, puede emplearse para expresar consultas de diversas fuentes de información, almacenadas o representadas como RDF.

- Protégé

  Protégé es una plataforma gratuita de código abierto que provee a la comunidad una suite de herramientas que facilitan la construcción de modelos de conocimiento y de aplicaciones basadas en conocimiento con el uso de ontologías [STANFORD, 2013].

  Entre sus características clave, Protégé implementa un conjunto de estructuras que modelan conocimiento y soportan la creación, visualización y manipulación de ontologías para que puedan ser representadas en diversas formas. Asimismo,

Protégé puede ser configurado para proveer un soporte amigable en la creación de modelos de conocimiento de un dominio y el ingreso de datos que esto requiere. Además, puede ser extendido, a través de un *plugin* y un API basado en Java, para construir otras herramientas y aplicaciones que se basan en conocimiento [STANFORD, 2013].

- Lucene

  Apache Lucene es una librería de código abierto que brinda la función de motor de búsqueda de textos y facilita la implementación de motores de búsqueda personalizados según las características propias de cada proyecto [APACHE, 2011].

  Entre algunas de las características que ofrece se encuentran:
  - Selección de tipos de consultas: por frases, proximidad, rango, etc.
  - Búsquedas por campos: título de documento, autor, contenidos.
  - Ordenación por campos.
  - Ranking de resultados

- Freeling

  Freeling es una suite de código abierto que provee servicios de análisis de lenguaje y que actualmente soporta el análisis para el idioma español [MAMBO, 2004].

2.3 Métodos y Procedimientos

Para la solución de este proyecto se empleará la técnica de expansión de consultas (ver sección 2.3.8) para enriquecer los términos de la consulta inicial formulada por el usuario. La forma elegida para realizar la expansión de consultas será automática y sin asistencia del usuario.

Como se mencionó en el párrafo anterior, la expansión de consultas permite añadir términos significativos a la consulta. De esta forma, los nuevos términos proveen información contextual que permite mejorar los resultados de la búsqueda. Esta información contextual puede obtenerse por distintas formas como *feedback* de relevancia, co-ocurrencia de términos o derivarse de modelos de conocimiento como ontologías.

Feedback de relevancia es una técnica que consiste en modificar la consulta inicial empleando palabras de los documentos mejor 'rankeados' o identificados como más relevantes. Para lograr esto, esta técnica requiere un ciclo en el que inicialmente el usuario recibe un conjunto de documentos como resultado de la búsqueda y hace una selección inicial de los documentos que él considera relevantes, y en base a este *feedback* se realiza la expansión que genera nuevos términos para añadir a la consulta inicial [BHOGAL, 2007].

La técnica de expansión basada en co-ocurrencia de términos hace referencia a la generación de los nuevos términos basado en modelos que contienen las frecuencias de ocurrencia de algunos términos junto (cerca) a otros y, por tanto, la inferencia de los nuevos términos se realiza en base a probabilidades [BHOGAL, 2007].

Para este proyecto se escogió aplicar la técnica de expansión basada en ontologías de dominio. En un lenguaje natural, una palabra puede tener múltiples significados dependiendo del contexto, por lo que el propósito de la ontología será añadir nuevos términos que permitan proveer un contexto más preciso para la búsqueda y así obtener como resultado mejoras en la relevancia de los documentos recuperados, mejoras medidas en términos de precisión y *recall* [BHOGAL, 2007].

2.4 Metodologías

Para el proyecto:
Para la revisión del estado del arte se realizó una revisión sistemática de la literatura.

2.4.1 Revisión sistemática

Una revisión sistemática de la literatura es una forma de identificar, evaluar e interpretar toda la investigación disponible relevante a cierta pregunta de investigación en particular, tema o fenómeno de interés [KITCHENHAM, 2004].

Algunas de las razones más comunes para realizar una revisión sistemática son [KITCHENHAM, 2004]:

- Resumir la evidencia existente relacionada a una tecnología
- Identificar vacíos en la investigación actual con la finalidad de sugerir áreas para desarrollar en trabajos futuros.

- Proveer un *framework/background* que facilite el posicionamiento de nuevas actividades de investigación.

Algunas de las características que las diferencian de las revisiones convencionales de literatura son [KITCHENHAM, 2004]:

- Empiezan definiendo un protocolo de revisión que especifica la pregunta principal y los métodos que se usarán para realizar la revisión.
- Se basan en estrategias definidas que buscan detectar la mayor cantidad de literatura relevante posible.
- Requieren criterios específicos de inclusión y exclusión para asegurar cada fuente de estudio potencial.
- Especifican la información que se obtendrá de cada fuente, incluyendo los criterios de calidad con los que se evaluarán los estudios.

Las fases de una revisión sistemática son: Planificación de la revisión, conducción de la revisión y reporte de la revisión.

Para este trabajo en particular se siguió al detalle la segunda fase según las siguientes actividades [KITCHENHAM, 2004]:

1. Identificación de la investigación.
2. Selección de los estudios primarios.
3. Estudio del aseguramiento de la calidad.
4. Extracción de información y monitoreo.
5. Síntesis de información.

3 Alcance

El proyecto a desarrollar pertenece al área de ciencias de la computación, y en particular al sub-área de ingeniería del conocimiento.

Para este proyecto se eligió diseñar un **modelo** en vez de crear un software (motor de recuperación) ya que el diseño del modelo permite enfocarse más en la definición de los mecanismos de recuperación y de la interacción de los componentes que en el proceso de desarrollo del software en sí. Asimismo, el diseño del modelo permite que el mismo pueda ser implementado utilizando herramientas, independientemente de la plataforma tecnológica.

Por otro lado, con la finalidad de probar el modelo se utilizarán herramientas ya existentes que faciliten la configuración de un motor de recuperación que opere bajo el modelo lógico propuesto.

Para este proyecto, se ha excluido del alcance los procedimientos específicos relacionados a la selección y extracción automática de información de fuentes *online*, y en su lugar se utilizará información específica ingresada manualmente a la herramienta por ser considerado suficiente para realizar las pruebas de los entregables acorde al modelo diseñado.

De forma análoga, el alcance excluye la creación de un mecanismo de captura de información del usuario, por considerarse un tema amplio que podría desarrollarse en trabajos futuros. Por esa razón, para las pruebas de los entregables se empleará información (conocimiento) codificada de un usuario manualmente ingresada (información del usuario no extraída automáticamente).

Finalmente, se ha excluido del alcance el análisis y pre-procesamiento de la consulta inicial del usuario por lógica proposicional, por considerarse un tema extenso que va más allá de la problemática inicial planteada para este proyecto, no obstante, se propone su desarrollo en futuros trabajos.

3.1 Riesgos

A continuación se muestra una tabla con los principales riesgos identificados de este proyecto de tesis, así como su impacto y las medidas que se tomarán para mitigarlos.

Riesgo identificado	Impacto en el proyecto	Medidas correctivas para mitigar
Alta complejidad en las herramientas a utilizar para el desarrollo de los entregables genere retrasos en el cronograma establecido de entregas.	Medio	- Revisión y capacitación anticipada de las herramientas a emplear para el desarrollo de los entregables con tutoriales de Internet, manuales. - Identificación de personas con dominio en dichas herramientas y consulta anticipada de dudas que pudieran obstaculizar los avances.
Alta complejidad en las herramientas a utilizar para el	Medio	- Revisión y capacitación anticipada de las herramientas a emplear para

desarrollo de los entregables genere presentación de entregables incompletos o de baja calidad.		el desarrollo de los entregables con tutoriales de Internet, manuales. - Identificación de personas con dominio en dichas herramientas y consulta anticipada de dudas que pudieran obstaculizar los avances.
Alta complejidad en las herramientas a utilizar para el desarrollo de los entregables retrase la entrega dentro de lo planificado.	Medio	- Revisión y capacitación anticipada de las herramientas a emplear para el desarrollo de los entregables con tutoriales de Internet, manuales. - Identificación de personas con dominio en dichas herramientas y consulta anticipada de dudas que pudieran obstaculizar los avances.
Retrasos en la entrega de avances parciales al asesor genere un incumplimiento del cronograma.	Medio	- Definir un cronograma de revisiones parciales que represente un compromiso de entrega al asesor.
Retrasos en la entrega de avances parciales al asesor genere una baja calidad de los entregables.	Medio	- Definir un cronograma de revisiones parciales que represente un compromiso de entrega al asesor.
Retrasos en la entrega de avances parciales al asesor genere que los entregables completados sin revisar no correspondan a lo necesario para que se cumpla el objetivo específico.	Medio	- Definir un cronograma de revisiones parciales que represente un compromiso de entrega al asesor.
Falta de ontologías del dominio elegido impidan el desarrollo del proyecto de tesis.	Alto	- Verificar la disponibilidad de ontologías del dominio elegido antes de iniciar la elaboración de la solución.
Actualizaciones imprevistas de las librerías o herramientas a usar requieran modificar lo avanzado hasta el momento para los entregables y genere retrasos.	Bajo	- Verificar continua y anticipadamente la evolución de las versiones de las herramientas.

Tabla 2. Riesgos identificados en el proyecto.

4 Justificativa y viabilidad del proyecto

4.1 Justificativa

La solución propuesta, como se ha planteado a través de sus entregables, busca presentar un modelo que permita la recuperación de información relevante con un especial énfasis en las búsquedas en el dominio de las ciencias de la computación. Esta orientación es particularmente conveniente y de utilidad para fines académicos, ya que el resultado del modelo desembocado en una aplicación constituye una herramienta de gran utilidad para que estudiantes de carreras afines a ingeniería informática y ciencias de la computación recuperen información relevante para sus estudios o proyectos de investigación.

Asimismo, este modelo no está restringido a funcionar únicamente en búsquedas en el dominio seleccionado, ya que al cambiar la ontología utilizada como base de conocimiento, sigue siendo factible emplear este modelo para realizar búsquedas en dominios propios de otras áreas de ciencias o ingenierías, ampliando así la cobertura de beneficiarios a estudiantes de otras especialidades.

En tercer lugar, el resultado de este trabajo es un modelo conceptual que tiene la ventaja de poder ser implementado sobre distintas plataformas e independiente de la tecnología usada para su implementación.

Finalmente, este modelo servirá de base para trabajos futuros en los que se desarrollará los aspectos que fueron excluidos del alcance de este proyecto o que no se profundizaron por no ser parte del objetivo principal de este trabajo. Entre ellos se puede mencionar los procedimientos específicos relacionados a la selección y extracción automática de información de fuentes *online* o los mecanismos de extracción automática de información del usuario alrededor del uso de estructuras ontológicas.

4.2 Viabilidad

4.2.1 Viabilidad técnica:

Todas las herramientas necesarias para la elaboración de los entregables son gratuitas y de código abierto, por lo que no habrán mayores limitantes o restricciones para su utilización. Asimismo, todas las librerías o APIs cuentan con versiones compatibles con el lenguaje de programación elegido (Java) por lo que no habrá dificultades en que interactúen en un mismo proyecto.

Por otro lado, para este trabajo se cuenta con el apoyo de un grupo de investigación cuyos miembros tienen conocimientos técnicos de estas herramientas puesto que también se encuentran desarrollando proyectos alrededor de temas de ingeniería del conocimiento o afines.

Por lo anteriormente expuesto se concluye que este proyecto es técnicamente viable.

4.2.2 Viabilidad temporal:

A continuación se muestra una tabla con la estimación del tiempo requerido para el desarrollo de cada uno de los resultados.

Resultado esperado	# de semanas
RE1 y RE2	2
RE3 y RE4	3
RE5	2
RE6 y RE7	3
RE8 y RE9	3
Total	**13**

Tabla 3. Estimación del tiempo requerido para el desarrollo de los resultados.

Según lo estimado, ya que el desarrollo de los entregables se completará en 13 semanas y dentro del plazo delimitado por el curso de proyecto de tesis 2, se concluye que este proyecto es temporalmente viable.

4.2.3 Viabilidad económica:

Para el desarrollo de este trabajo ya se dispone de una computadora portátil y otra de escritorio. Asimismo, este proyecto no requiere de mayor inversión económica ya que las herramientas a utilizar son gratuitas y el material para la investigación del tema es de acceso libre (no restringido y gratuito) dentro del campus universitario y de la sala facilitada por el grupo de investigación. Por lo tanto, se concluye que este proyecto es económicamente viable.

4.2.4 Análisis de necesidades:

Este proyecto se desenvuelve principalmente dentro del ámbito de estudio de la ingeniería del conocimiento debido a su fuerte interacción con las ontologías. Sin embargo, también consta de una parte menor que interactúa con el área de

procesamiento del lenguaje natural, según lo especificado en los resultados esperados. Para ambos casos se cuenta con el apoyo de dos miembros del grupo de investigación anteriormente mencionado, el primero, doctor en ingeniería del conocimiento y el segundo, especialista en procesamiento del lenguaje natural quienes proveerán su apoyo y orientación en todas las fases del desarrollo de este proyecto.

CAPÍTULO 2

1 Marco conceptual

1.1 Introducción

A continuación se presentará algunos conceptos de importancia para la comprensión de este proyecto. Se iniciará con una breve definición de recuperación de información, seguido de los principales conceptos relacionados al funcionamiento de los sistemas IR, desde características generales de estos sistemas hasta características particulares que hacen diferente la forma de procesamiento las búsquedas y por ende los resultados.

Lo que se busca a través del desarrollo del marco conceptual es brindar los conceptos necesarios para comprender tanto la problemática central de este proyecto y el entorno en el que se desenvuelve, así como los conceptos que permitirán entender el enfoque al que se orientará el desarrollo de este proyecto en los siguientes capítulos.

1.2 Recuperación de información

Algunos términos clave que se ha logrado identificar de las definiciones de recuperación de información brindadas por distintos autores y que nos brindan un panorama inicial antes de proceder con una definición formal, han sido agrupadas según la pregunta que podría estar respondiendo dicho concepto. Esto se ha resumido y se puede observar en la tabla 4.

Pregunta que podría responder el concepto	Concepto	Autores
¿Cuál es la motivación (disparador)?	• Un usuario tiene una necesidad de información / requerimiento de información / una pregunta / una consulta.	• Rijsbergen, 1979. • Ingwersen, 1992.
¿Qué necesita obtener el usuario?	• Necesita obtener información relevante / información útil plasmada en documentos.	• Rijsbergen, 1979. • Ingwersen, 1992. • Baeza Yates and Riberiro Neto, 1999. • Mitra and • Chaudhuri, 2000.
¿De dónde se busca obtener	• Se busca obtener de	• Greengrass, 2000.

esa información?	documentos de un repositorio / colección / archivo / biblioteca digital.	• Mitra and Chaudhuri, 2000
¿Qué factores (restricciones) se debe considerar?	1. Se quiere tener un acceso rápido a la información. 2. La información se encuentra sin estructurar / los documentos o textos están en lenguaje natural sin limitaciones.	• Baeza Yates and Riberiro Neto, 1999. • Greengrass, 2000. • Mitra and Chaudhuri 2000 • Cerulo and Canfora 2004.
¿Qué hace la disciplina de recuperación de información?	• Diseñar, construir y probar sistemas de recuperación de información. • Realiza el análisis, diseño e implementación de sistemas computarizados de recuperación de información.	• Ingwersen, 1992. • Cerulo and Canfora, 2004.
¿Con qué procesos trata?	• Con la representación, almacenamiento, organización y acceso a elementos de información. • Con la representación, almacenamiento, búsqueda y recuperación de información.	• Baeza Yates and Riberiro Neto, 1999. • Cerulo and Canfora 2004

Tabla 4. Identificación de conceptos comunes en las definiciones de recuperación de información de diversos autores.

De una manera simple, la disciplina de la recuperación de información busca, a través de sistemas computarizados, encontrar documentos que puedan potencialmente satisfacer una necesidad de información. Esta necesidad de información es expresada por un *query* generado por el usuario [GREENGRASS, 2000].

Finalmente, derivado de los aportes en este ámbito de los mencionados autores se puede más formalmente entender a la recuperación de información como la disciplina científica que se encarga del análisis, diseño e implementación de sistemas

computarizados que tratan con la representación, almacenamiento, organización y acceso a información sin estructurar la cual da respuesta a la consulta de un usuario.

1.2.1 Concepto de relevancia y no relevancia

A los documentos que satisfacen la consulta desde el punto de vista del usuario se les dice que son "relevantes" y los que no, son "no relevantes" [GREENGRASS, 2000].
De esta forma, la disciplina de recuperación de información se orienta a maximizar la devolución de documentos relevantes, y minimizar la devolución de documentos no relevantes, lo cual se podrá medir a través de los conceptos de precisión y *recall* (ver sección 2.3.4).

1.2.2 Recuperación de datos vs. Recuperación de información

La recuperación de datos, en el contexto de un sistema IR, consiste principalmente en determinar qué documentos de una colección contienen las palabras clave de la consulta del usuario, lo cual mayormente no es suficiente para satisfacer las necesidades de información del usuario. En contraste, el usuario de un sistema IR está más interesado en recuperar información acerca de un tema, más que recuperar data que satisface una consulta [BAEZA-YATES, RIBERIRO-NETO 1999].

Algunas de las propiedades distintivas entre recuperación de datos y recuperación de información se puede apreciar en la siguiente tabla:

Propiedad	Recuperación de datos	Recuperación de información
Matching (Coincidencia)	Coincidencia exacta	Coincidencia parcial
Lenguaje de la consulta	Artificial	Natural
Items deseados	Que coincida (con los términos del query)	Relevantes
Respuesta ante el error	Sensible	Insensible

Tabla 5. Comparación de recuperación de datos vs. Recuperación de información.
Tabla adaptada de [RIJSBERGEN, 1979].

En el ámbito de recuperación de datos, normalmente se busca una correspondencia exacta, es decir, se verifica si un término está o no presente en un documento.

En cambio, en el caso de la recuperación de información, si bien es cierto que la existencia de una correspondencia exacta pudiera ser de interés, generalmente lo que se quiere encontrar son los elementos de información que parcialmente satisfagan la consulta o solicitud, y entre ellos seleccionar a los mejores previo ordenamiento bajo cierto criterio de relevancia acorde a la consulta del usuario [RIJSBERGEN, 1979].

Asimismo, a diferencia de los sistemas de recuperación de datos, para un sistema IR es aceptable que los objetos recuperados puedan ser no precisos. Es decir, un documento puede ser considerado como aceptable o relevante aunque no necesariamente haya habido una correspondencia plena entre los términos de la consulta y de dicho documento. La razón principal para esta diferencia es que la recuperación de información usualmente trata con texto en lenguaje natural, el cual no siempre está bien estructurado y puede ser semánticamente ambiguo. Mientras que un sistema de recuperación de datos (como una BD relacional) trata con data que tiene una estructura y semántica bien definida [BAEZA-YATES, RIBERIRO-NETO 1999].

1.2.3 Componentes típicos de un sistema IR

Tanto Rijsbergen como Ingwersen han identificado de manera similar (si bien con distintos nombres) a los componentes de un sistema de recuperación de información básico como se indica a continuación.

Para Rijsbergen los componentes son: *input*, procesador y *output*.
En primer lugar, del lado del *input*, este está formado por una representación apropiada de cada documento y la consulta. Una representación puede ser por ejemplo, una lista de palabras consideradas 'significativas'. En segundo lugar, el procesador es la parte del sistema que se encarga del proceso de recuperación. Este proceso puede involucrar realizar la función de recuperación, la cual consiste en ejecutar una estrategia de búsqueda en respuesta a la formulación de una consulta. Finalmente se obtiene el *output*, el cual normalmente es un conjunto de citas o códigos de documentos [RIJSBERGEN, 1979].

Los componentes según Ingwersen son la información potencial, las entidades de texto devueltas y la función de *match*. Entre estos componentes y los planteados por Rijsbergen (*input*, *output* y procesador) puede observarse una correspondencia respectivamente [INGWERSEN, 1992].

1.2.4 Medición de la efectividad de un sistema IR

La efectividad (o performance) se mide en términos de precisión y *recall*. Se considera que los de mayor efectividad son los que llevan a altos niveles de precisión así como altos niveles de *recall* en simultáneo [SALTON, 1981].

1.2.4.1 Precisión

Precisión es la habilidad de rechazar elementos no pertinentes que no son deseados [SALTON, 1981]. Formalmente se puede expresar como se indica a continuación [RIJSBERGEN, 1979]:

$$Precision = \frac{\#\ de\ documentos\ relevantes\ recuperados}{\#\ total\ de\ documentos\ recuperados}$$

1.2.4.2 Recall

Recall es la habilidad del sistema de recuperar elementos deseados por los usuarios [SALTON, 1981]. Formalmente se puede expresar como se indica a continuación [RIJSBERGEN, 1979]:

$$Recall = \frac{\#\ de\ documentos\ relevantes\ recuperados}{\#\ total\ de\ documentos\ relevantes\ (recuperados\ o\ no)}$$

1.2.5 Representación de los documentos almacenados en un sistema IR

Los documentos en una colección a menudo se representan a través de un conjunto de términos índices o palabras clave. Esas palabras clave pueden ser extraídas directamente del texto del documento o pueden ser especificadas por una persona [RIJSBERGEN, 1979]. Asimismo, sin importar si estas palabras representativas se generan automáticamente o son generadas por un especialista, cumplen con proveer una vista lógica del documento [BAEZA-YATES, RIBERIRO-NETO 1999].

1.2.5.1 Indexación

A este proceso de asignar los términos o índices a cada documento (que en conjunto representan el contenido del documento) se le conoce como indexación. La tarea de indexar es crucial para la recuperación de información, ya que fallos en la política de indexación llevan inmediatamente a fallos de la recuperación. Por ejemplo, si la indexación no es lo suficientemente exhaustiva, es decir, si los términos no reflejan apropiadamente el contenido del documento, podría ser que dicho documento no fuera recuperado cuando se necesite. Por otro lado, si los términos

asignados son muy extensos y muy poco específicos, podría ser que no se rechace un documento que claramente no es relevante [SALTON, 1981].

1.2.5.2 Frecuencia de un término

La frecuencia de un término de un documento se define como la frecuencia de ocurrencia (aparición) de dicho término en un documento dado, mientras que la frecuencia de un término de una colección es el número de documentos en una colección en los que ese término aparece. Por esa razón, los mejores términos tendrán una alta frecuencia del término en documentos individuales, pero una baja frecuencia en la colección [SALTON, 1981].

1.2.5.3 Frecuencia invertida de un término

Dada una colección de muestra de documentos o fragmentos de documentos relacionados a un tema, es posible introducir una función de ponderación **IDFij** basada en frecuencias, que refleja la importancia del término Tj para la representación del contenido del documento Di. A esa función se le conoce como la frecuencia invertida del documento y su expresión es como se indica a continuación:

$$IDFij = \frac{Frecuencia\ de\ Tj\ en\ Di}{Frecuencia\ de\ colección\ de\ Tj}$$

Esta función provee la base para una estrategia de indexación automática usando términos (índices) extraídos de extractos de documentos o *abstracts* [SALTON, 1981].

1.2.6 Proceso de recuperación de información

Partiendo de la premisa de que se cuenta con una colección de documentos (textos) los cuales ya han pasado por el proceso de indexación descritos en el punto anterior, se puede asumir entonces que ya se cuenta con una vista lógica de cada documento de la colección. A partir de ahí el proceso de recuperación de información es de la siguiente forma:

Primero, el usuario especifica una necesidad suya la cual es transformada por las mismas operaciones que se le aplicaron a los documentos de la colección. Luego, el *query* es procesado para obtener como resultado documentos. Un procesamiento rápido de la consulta se logra gracias a la estructura de índices construida anteriormente. Antes de ser enviados al usuario, los documentos recuperados son reordenados (ranking) según su nivel de relevancia. A continuación, el usuario examina el conjunto de documentos –ya priorizados– en búsqueda de información útil. En este punto él podría señalar un subconjunto de estos documentos como

definitivamente de interés para él, y con esto se iniciaría un ciclo de *user feedback* (retroalimentación del usuario). En ese ciclo, el sistema usa los documentos seleccionados por el usuario para cambiar la formulación del *query*. Se espera que este *query* modificado sea una mejor representación de la verdadera necesidad del usuario [BAEZA-YATES, RIBERIRO-NETO 1999].

1.2.7 Modelos de sistemas IR

Entre los principales modelos se encuentran:

1.2.7.1 Modelo Booleano

En el modelo booleano, la consulta es formulada como una combinación de términos. Una consulta booleana convencional usa los operadores clásico AND, OR, y NOT. Por ejemplo, la consulta "t1 AND t2" es satisfecha por un documento D1 si y solo si D1 contiene ambos términos. Esa consulta booleana clásica es o verdadera o falsa, de tal forma que un documento que satisface la consulta es *relevante* o si no lo satisface es *no relevante* [HARMAN apud GREENGRASS, 2000, p.13].

1.2.7.2 Modelo Vectorial

En el modelo vectorial podemos aplicar el proceso de indexación a cada documento de una colección, generando (mayormente bajo el esquema de frecuencia invertida de un término) un conjunto de términos que representan dicho documento. Si luego unimos todos esos conjuntos de términos, obtenemos un conjunto de términos que representan a toda la colección, conjunto al cual definimos como “espacio”.

Para poder encontrar un documento relevante a cierto *query* es necesario representar el *query* de la misma forma (mismo esquema de indexación) en que se generó el espacio de términos de la colección. Entonces, si definimos al vector $\vec{Q}$ como el vector de los pesos de cada término índice del *query* y al vector $\overrightarrow{D_i}$ como el vector de los pesos de cada término índice del documento, la similitud entre un *query* $\vec{Q}$ y un documento $\overrightarrow{D_i}$ se calcula como el coseno de ambos vectores normalizados [GREENGRASS, 2000; PARALIC, KOSTIAL 2003].

La expresión formal de esta similitud es como se indica a continuación:

$$sim_{TF-IDF}(\vec{Q}, \overrightarrow{D_i}) = \frac{\overrightarrow{D_i} x \vec{Q}}{|\overrightarrow{D_i}||\vec{Q}|}$$

1.2.8 Expansión de consultas

Usualmente los usuarios tienden a formular consultas muy cortas en vez de consultas largas y cuidadosamente armadas. Tales consultas cortas carecen de palabras que, si fueran provistas por el usuario, podrían ser términos de búsqueda útiles. Para poder alcanzar una efectividad razonable en la recuperación al procesar estas consultas cortas se hace importante contar con técnicas especiales [MITRA, SINGHAL et al 1998].

Debido a la ambigüedad del lenguaje natural y también a la dificultad de usar un solo término para representar un concepto de información, la expansión de consulta representa una buena alternativa.

Con la expansión de consultas, el usuario es guiado a formular consultas las cuales permiten obtener resultados útiles. El objetivo principal de la expansión de consultas es añadir nuevos términos significativos a la consulta inicial. Este proceso de añadir términos puede ser manual, automático o asistido por el usuario [BHOGAL, MACFARLANE et al 2007].

1.2.9 Ontologías

Las ontologías son una conceptualización de dominios (áreas) de conocimiento. Una ontología describe conceptos y relaciones que son importantes en un dominio particular y brinda un vocabulario para ese dominio así como una especificación (comprensible para los humanos y legible para las máquinas) del significado de los términos usados en ese vocabulario. En los últimos años, las ontologías han sido adoptadas en muchos negocios y comunidades científicas como una forma de compartir, reusar y procesar conocimiento de un dominio. Las ontologías actualmente tienen un rol central en muchas aplicaciones, tales como portales de conocimiento científico, sistemas de gestión de información, comercio electrónico y servicios de web semántica. [WANG, et al 2012; GENESERETH & NILSSON apud GRUBBER, 1995; STANFORD, 2013].

1.2.9.1 Ontologías en IR

Para un sistema de recuperación de información basado en ontologías, cuando el usuario ingresa una consulta, el sistema trata de insertar el conocimiento de las ontologías para enriquecer la expresión de la consulta y mejorar la probabilidad de obtener documentos más relevantes [WANG, et al 2012].

Esta es en esencia la principal aproximación de las ontologías en los sistemas IR. Distintos autores le han dado diferentes enfoques a la aplicación de ontologías en estos sistemas, enfoques los cuales se verán con mayor detalle en el apartado siguiente de estado del arte.

1.3 Conclusión

Como se ha podido observar a través de los conceptos expuestos anteriormente, el tema de la recuperación de información ha sido ampliamente investigado y reconocido como un tema de importancia por diversos autores, y en gran parte porque proviene de una necesidad elemental de los seres humanos como la continua búsqueda de conocimiento.

En este apartado se han revisado tanto los conceptos básicos involucrados en la disciplina de la recuperación de información, por ser el área central que rodea a la problemática, así como los conceptos relacionados al enfoque bajo el cual se desarrollará, para el dominio específico indicado en la problemática, una solución alternativa y que se verifique eficiente.

2 Estado del arte

2.1 Introducción

En los últimos años, la necesidad de compartir conocimiento entre diferentes aplicaciones dio pase a un creciente aumento de interés en la investigación de las ontologías. Asimismo, este hecho, junto a los avances en el mejoramiento de soluciones por la estrategia de expansión de consultas en IR, ha sido adecuadamente aprovechado por los investigadores quienes han brindado propuestas que integran ambos conceptos y han logrado soluciones que han demostrado significativas mejoras en cuanto a resultados.

El objetivo de la revisión del estado del arte es dar a conocer las diferentes propuestas que se han planteado a la fecha acerca del uso de las ontologías en el proceso de expansión de consultas como respuesta al problema identificado.

2.2 Método usado en la revisión del estado del arte

El método empleado para la revisión del estado del arte ha sido una revisión sistemática de la bibliografía actual existente.

Como parte de la revisión se tomaron las siguientes consideraciones:

- La pregunta de investigación se definió como "¿Cómo han sido utilizadas las ontologías en el proceso de expansión de consultas?".
- La lista de términos de búsqueda incluirían "*Query expansion*" y "*Ontology*" enlazados por el operador "*and*" y se fijaría la búsqueda de estos términos en los títulos de los documentos.
- Las fuentes seleccionadas para la ejecución de la búsqueda fueron *ACM digital library*, *IEEE Xplore Digital Library* y *Science Direct*.
- La selección de estudios se realizó a través de un proceso iterativo e incremental.
- Los criterios de inclusión fueron análisis del *abstract*, el título y las palabras clave (*keywords*) indexadas.
- El criterio de exclusión consistió en restringir la búsqueda a las publicaciones dentro del rango de años comprendido entre el 2003 (inclusive) y la actualidad.
- Protocolo de revisión:
    1. Artículo
    2. Sistema
    3. Dominio de aplicación
    4. Métodos, procedimientos, herramientas usadas

5. Idioma
6. Algoritmos
7. Mecanismos de validación
8. Modelo, arquitectura propuesta
9. Tipo de información recuperada en el sistema IR
10. Año de publicación

2.3 Aplicación de ontologías independientes de un dominio o generales

Las ontologías han sido utilizadas para asistir al proceso de expansión de consultas desde principios de los 90 con buenos resultados.
Asimismo, *WordNet* ha sido una ontología general popular usada en el área de expansión de consultas desde aquel entonces [BHOGAL, MACFARLANE et al 2007].

En el 2003, Navigli y Velardi emplearon las ontologías no sólo expandiendo las consultas con sinónimos e hiperónimos, ya que consideraban que éstos tienen un efecto limitado en el desempeño de los sistemas de información web. Ellos sugirieron que otros tipos de información derivables de una ontología tales como "nodos comunes" son más efectivos. Esto ya que consideran que palabras en el mismo dominio semántico y con el mismo nivel de generalidad son mejores candidatos para la expansión.
Bajo su enfoque, la ontología es usada para extraer el dominio semántico de la palabra y luego la consulta es expandida usando co-ocurrencia de palabras.
Sus experimentos usaron *WordNet* para la ontología y Google para el proceso de recuperación. Para estos experimentos, crearon una red semántica para cada 'sentido' de las palabras. Luego las redes semánticas relevantes se intersectaban y se asignaba un puntaje basado en el número de nodos comunes donde nodos comunes son aquellos que podían alcanzarse por ambas redes semánticas por rutas directas. Los resultados de sus experimentos mostraron una mejora con respecto a la consulta sin expandir [NAVIGLI, BELARDI apud BHOGAL, 2007].

2.4 Aplicación de ontologías de dominio

El problema con las ontologías independientes de un dominio como *WordNet* es que ya que tienen una amplia cobertura, la ambigüedad de los términos dentro de dicha ontología puede resultar problemática. En cambio, para tareas de búsqueda con una orientación directa a un tema, son preferidas las ontologías específicas a un dominio. Una ontología específica de dominio modela términos y conceptos que son específicamente usados en dicho dominio. Asimismo, muchas ontologías específicas

de un dominio se han construido para muchas áreas de aplicación distintas tales como medicina, leyes, agricultura, geografía, multimedia, negocios entre muchos otros [BHOGAL, MACFARLANE et al 2007].

En el 2008, Jalali y Borujerdi estuvieron de acuerdo en que no es óptimo emplear ontologías de propósito general como *WordNet* para dominios específicos ya que pueden causar la pérdida de precisión en la recuperación de información.
Ellos consideraron que utilizar diversas ontologías de dominio ya existentes podría ser más beneficioso para evitar los dilemas de construir una ontología, además de la posibilidad de explotar el conocimiento auxiliar que brindan las estructuras de conocimiento externas que no pueden ser fácilmente inferidas. Por ello presentaron un algoritmo de expansión de consultas para la recuperación de información médica.
Su propuesta consiste en identificar conceptos de la ontología MeSH (*Medical Subject Headings*) en la consulta de los usuarios aplicando un algoritmo de expansión a dichos conceptos. Este algoritmo de expansión estaría basado en el número de sinónimos de cada concepto, el número de términos que conforman el concepto y la ubicación de los conceptos en la jerarquía del MeSH.

Los resultados mostraron mejoras sobre la expansión usando ontologías de propósito general y otras aproximaciones [JALALI, BORUJERDI 2008].

En el 2005, Dey y Singh propusieron un sistema basado en un mecanismo de expansión de consultas en el que para realizar la expansión se emplea el conocimiento ontológico de una o más ontologías. Es decir, usan la técnica de expansión de consultas para combinar conceptos de más de una ontología de dominio. Esto fue logrado al mezclar los espacios de conceptos de diferentes ontologías junto con conceptos terminológicamente relacionados del concepto original de la consulta. Para comprender mejor este concepto nos referiremos al ejemplo indicado en su propuesta.
Dada la consulta: “parte de la Alfalfa afectada por el moho”, por un lado, el concepto de que “el moho es una especie de enfermedad de las plantas” es obtenido de la ontología de las enfermedades de las plantas, la cual almacena información acerca de partes de las plantas afectadas por varias enfermedades. Por otro lado, “la alfalfa es una clase de legumbre” se obtiene de una ontología de taxonomía de las plantas. De esta manera, haciendo uso de ambas ontologías y la técnica de la expansión, observaron que se puede obtener mejores resultados. Esto también fue corroborado en los resultados de sus pruebas, los cuales indicaron que la precisión de la

recuperación es mucho mejor para la consulta transformada que la original [DEY, et al 2005].

Otro de los enfoques es propuesto por Wang quien en su trabajo introduce las ontologías en el proceso de expansión haciendo buen uso de las relaciones semánticas de los conceptos de dichas ontologías para expandir las palabras clave (*keywords*) de las consultas. En dicho trabajo se señala que, partiendo de la arquitectura propuesta, uno de los módulos encargado de la expansión realizaría este proceso –de expansión– sobre las palabras clave (*keywords*) de la consulta basándose en las relaciones y en mecanismos de razonamiento de la ontología. Este módulo emplea un algoritmo para extraer la relación de sinónimos o relaciones de padre, hijo y hermano en las ontologías, y genera palabras previas a la expansión, las cuales son reordenadas (*ranking*) y con esas palabras genera una nueva expresión de consulta, la cual posteriormente es enviada y procesada por el módulo de recuperación de información. Los resultados evidenciaron mejoras en los ratios de precisión y *recall* del sistema [WANG, et al 2009].

Ali y Khan por su parte, a través de una propuesta de aplicación de expansión de consultas en integración de datos, tienen como objetivo extender los resultados de determinada consulta en una forma semánticamente significativa que capture mejor las necesidades del usuario incluso si éste no tiene conocimiento del dominio o de la estructura interna de las fuentes de datos. En el modelo propuesto ellos se orientan a no solamente tener como fuente de conocimiento documentos sin estructurar. Ellos se proponen integrar distintas fuentes (estructuradas, semi-estructuradas y no estructuradas) y asociarlas con la ontología de dominio, la cual constituye la fuente del conocimiento semántico, y sobre esta unificación realizar la expansión de consultas [ALI, KHAN 2008].

2.5 Aplicación de Ontologías geográficas

El uso tradicional de las ontologías en la expansión de consultas es apropiado para procesar consultas generales y obtener mejorías en los resultados como se ha observado en párrafos anteriores, sin embargo no provee un soporte adecuado para procesar consultas espaciales [FU, et al 2005].

En el 2005, Fu consideró que dicho tipo de soporte es necesario ya que la mayor parte de actividades de los humanos parten de un espacio geográfico de alguna forma, y en consecuencia, muchos documentos incluyen referencias a contextos geográficos,

típicamente a través de nombre de lugares. Por ejemplo, una consulta a resolver podría ser “Castillos cerca de Edimburgo”, la cual involucra términos espaciales (como “Edimburgo”) y relaciones espaciales difusas (como “cerca”) que califican a los términos espaciales.

Ante esto, propone una técnica que la distingue de las convencionales ya que la consulta es expandida por derivación de su rastro (*footprint*) geográfico en ella, de tal forma que se obtengan documentos *espacialmente* relevantes. En la arquitectura de su sistema se incorpora un componente ontológico, cuyas partes principales son una ontología de dominio y una ontología geográfica (o geo-ontología). Por un lado, la ontología de dominio modela la terminología de un área de aplicación o dominio y se usa para resolver el aspecto del “qué” de la consulta, mientras que por otra parte, el aspecto del “dónde” es manejado por la geo-ontología, la cual contiene distintos tipos de información incluyendo los nombres con los que se le conoce a un lugar, los nombres de los tipos de lugar con los que puede estar categorizado, sus relaciones topológicas (como “parte de” y “contiene”) con otros lugares, entre otros.
Los resultados mostraron una considerable mejoría en los resultados cuando una consulta involucraba una relación espacial difusa [FU, et al 2005].

2.6 Aplicación de ontologías difusas

Recientemente la lógica difusa ha sido integrada a las ontologías para definir un nuevo paradigma teórico llamado ontología difusa. Este es un primer paso para evitar la vaguedad o generalidad característica del conocimiento representado en determinado dominio, además, al emplear una ontología difusa la fase de la expansión se realiza de mejor forma, ya que solo los padres, hijos e instancias que incluyen los términos de la consulta hasta cierto grado son considerados [CALEGARI, SANCHEZ 2008].

En el 2008, Calegari y Sanchez proponen una aproximación para mejorar la recuperación de documentos basada en ontologías difusas. En ella emplean un algoritmo usando una red conceptos de objetos difusos (O-FCN). Este algoritmo permite derivar un único camino entre entidades involucradas en la consulta para obtener las mejores asociaciones semánticas en el dominio de conocimiento [CALEGARI, SANCHEZ 2008].

Posteriormente en ese mismo año, Calegari y Pasi proponen una técnica de expansión de consultas a través de la definición de una ontología difusa personalizada que emplea las consultas pasadas de un usuario almacenadas en un perfil de usuario (con

ayuda de un *gadget* de Google llamado *Personal Information Context*). La expansión de consultas se realiza no sólo analizando las relaciones de correlación generadas por la red O-FCN anteriormente mencionada, sino también por los conceptos contenidos en documentos relevantes almacenados en la computadora del usuario. Asimismo, cada vez que el usuario interactúa con el *gadget* las relaciones de la ontología son actualizadas [CALEGARI, PASI 2008].

De forma similar, Sendhilkumar y Gheeta propusieron una ontología personalizada usada como perfil de usuario la cual fue definida analizando las acciones que un usuario realizaba sobre páginas web (como la transición del usuario de una página a otra, el tiempo de visita a la página, la selección de *links*, entre otros), de tal forma que pudiera ser usado para confirmar el contexto de la búsqueda [SENDHILKUMAR, GEETHA 2008].

2.7 Aplicación de ontologías de interés

Para resolver el problema de que los métodos de expansión a veces no logran reflejar apropiadamente la intención de los usuarios, Chen y Du proponen un modelo de expansión basado en ontologías de interés de los usuarios.

Para ello primero se centran en la construcción de una ontología de interés del usuario en determinado dominio (con la ayuda del editor de ontologías de código abierto Protege), y a continuación presentan un algoritmo para calcular la similitud entre conceptos.

La arquitectura de su modelo incluye un módulo de expansión de consultas el cual añade a la consulta original los conceptos con niveles de similitud mayores a un umbral en base a la ontología de interés construida, luego de lo cual la consulta queda transformada de tal forma que puede ser enviada al módulo de recuperación de información.

Los resultados experimentales probaron que esta aproximación puede expresar más claramente la intención del usuario ya que el modelo logró niveles más altos de precisión y *recall* que otros métodos de expansión que no emplean ontologías de interés [CHEN, et al 2012].

2.8 Ventajas y desventajas del uso de ontologías

El uso de ontologías en el proceso de expansión de consultas ofrece ventajas como su alta disponibilidad en la red. Asimismo, al incluir en ellas palabras como sustantivos propios (nombres de personas y de lugares) se facilita la obtención de conceptos semánticamente relacionados a éstos.

En tercer lugar, existen muchas herramientas de software para automatizar la creación y soportar la evolución de las ontologías; y finalmente, la mayoría de ontologías han sido definidas en lenguajes portables como XML, con lo que es posible utilizar características de éste tipo de lenguajes para manejar los cambios en la evolución de las ontologías [BUCKLAND apud BHOGAL, 2007].

Sin embargo, las ontologías no están libres de problemas.

El primero está relacionado con el desfase entre los términos de una consulta y los conceptos en la ontología, por lo que se debe realizar un proceso de mapeo para superar este problema.

En segundo lugar, si una ontología de un dominio particular no existe se debe realizar mucho esfuerzo para construir ésta ontología desde un punto de vista técnico o más aún, poder extraer el conocimiento de expertos en dicho dominio y hacerlo aterrizar bajo un consenso [BHOGAL, MACFARLANE et al 2007].

2.9 Conclusiones sobre el estado del arte

En este apartado se ha observado cómo se han utilizado los diferentes tipos de ontologías en el proceso de expansión de consultas para dar alternativas de solución adecuadas como las que busca la disciplina de recuperación de información.

Como se ha visto, las ontologías generales brindan muy buenos resultados, sobre todo cuando no se aplican en búsquedas de ámbitos específicos, y tienen la ventaja de que no requieren necesariamente la interacción activa del usuario en el proceso de expansión a diferencia de otros tipos de expansión de consultas.

Por otro lado, las ontologías de dominio han mostrado un buen desempeño para dar resultados más precisos en búsquedas de dominios específicos. Asimismo, se ha observado cómo los autores han sabido aprovechar la variedad de éstas para obtener aún mejores resultados al hacerlas interactuar o extraer conocimiento procesando ambas en conjunto, o incluso haciendo sistemas que utilizan estas ontologías de dominio de la mano con otros tipos de ontología como las geo-ontologías que ayudan a procesar de forma más efectiva consultas con referencias espaciales.

Asimismo, otros tipos de ontología se han aplicado, si bien no de forma tan directa como con las técnicas de *feedback* de relevancia, para involucrar al usuario en el proceso de expansión al extraer conocimiento acerca de él (su perfil o sus intereses) de tal forma que este conocimiento pueda ser utilizado para resultados más precisos.
En este aspecto, los autores han considerado la creación de un solo perfil de usuario que pudiera ir mejorándose, sin embargo, se considera que un concepto como lo planteado por Bai y Nie acerca del uso de varios perfiles de usuario dependiendo del contexto de la búsqueda podría ser aterrizado de forma efectiva también con el uso de ontologías [BAI, NIE 2007].

Las ontologías, como se ha visto, brindan una serie de ventajas pero también presentan algunas limitaciones. Sin embargo, se considera que estas limitaciones son ampliamente superables ya que con la propuesta de algoritmos apropiados para realizar el mapeo entre términos (de la consulta) y conceptos (de las ontologías), y de métodos para superar la carencia de ontologías para determinados dominios aprovechando ontologías de otros dominios como lo han demostrado los autores, se puede conseguir resultados positivos del uso de las ontologías que superan el esfuerzo requerido necesario para saltear estas limitantes.

CAPÍTULO 3

1 Objetivo específico 1: Diseñar un mecanismo que permita derivar la consulta del usuario en una consulta más específica en un dominio.

Gran parte de las fuentes de conocimiento que pueden ser consultadas y entregadas a los usuarios para satisfacer sus necesidades de información se encuentran escritas en lenguaje natural. Siendo así, es común encontrar que un mismo concepto está representado de distintas formas no sólo por la diversidad determinada por la existencia de palabras equivalentes o sinónimas respecto a un mismo concepto (la cual será considerada en un siguiente capítulo de forma independiente), sino también por la variedad de representaciones que pueden derivar de las reglas y principios sintácticos de cada lenguaje.

Teniendo en consideración dicha realidad, este objetivo está orientado a superar la dificultad determinada por la diferencia entre cómo está expresado el conocimiento en los documentos y cómo fue planteado en la consulta, considerando que ambos están representados en lenguaje natural con las características que ello conlleva.

Lo que se espera con el desarrollo de este objetivo es que el usuario pueda ingresar libremente una consulta sin necesidad de forzar sus palabras de búsqueda a vocablos en forma base o canónica, y que el hecho de que los conceptos de las fuentes de información se hallen representados de forma diversa no sea una limitante para la efectividad de la recuperación.

1.1 Resultado esperado 1: Mecanismo de pre-procesamiento de textos en lenguaje natural que remueva las palabras vacías como artículos y conectores (stopwords).

Este mecanismo de procesamiento del lenguaje natural permitirá remover de la consulta o de las palabras clave de las fuentes, aquellas que no brinden un aporte en la identificación de conceptos relevantes para el proceso de recuperación. Estas palabras son conocidas como *stopwords* e incluyen palabras vacías como artículos y conectores.

1.1.1 Pre-requisitos

Para el correcto funcionamiento de este mecanismo se debe contar con la consulta inicial del usuario como input, y con un diccionario de stopwords, es decir, una fuente

que contenga un conjunto de palabras que sean consideradas como vacías para el idioma español.

1.1.2 Funcionamiento

En primer lugar, la consulta inicial del usuario debe pasar por un proceso de tokenización que consiste en separar dicha consulta en un conjunto *tokens* (palabras) independizados.

A continuación, cada *token* es evaluado con ayuda de un componente que verifica si dicha palabra es o no una *stopword.* Esto se realiza haciendo la consulta contra un diccionario de *stopwords* en idioma español.

De darse el caso de que el *token* evaluado sí sea un *stopword*, dicha palabra ya no es incluida en la consulta resultante.

Finalmente, este mecanismo retorna la consulta pre-procesada, es decir, la consulta del usuario sin incluir los *stopwords* en español.

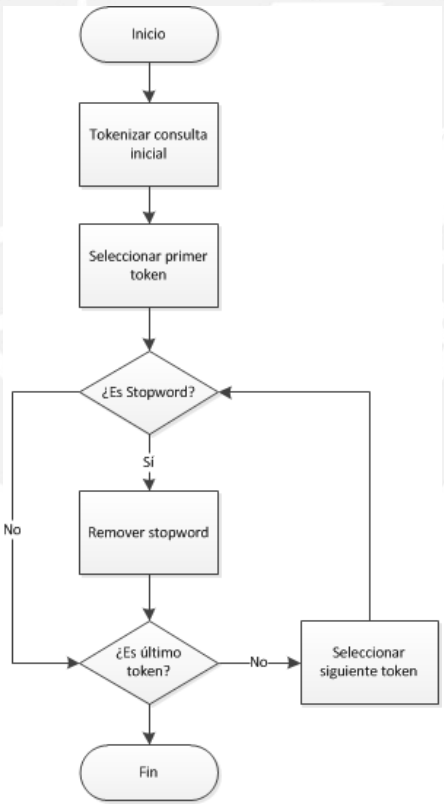


Imagen 2. Mecanismo de remoción de *stopwords*

1.1.3 Mecanismo de prueba

Para la verificación del mecanismo se realizó la siguiente prueba:

Input (Consulta inicial):

Consulta inicial: Uno de los objetivos de este resultado es eliminar palabras conocidas como stopwords que no son relevantes para el proceso de recuperación.

Output (Consulta sin *stopwords*):

Consulta sin stopwords: objetivos resultado eliminar palabras conocidas stopwords relevantes proceso recuperación

De estos resultados se puede observar que las palabras eliminadas por ser *stopwords* fueron:
“Uno de los de este es como que no son para el de”, lo cual es pertinente ya que dichas palabras no son relevantes y más bien representan ruido para el proceso de recuperación.

1.1.4 Consideraciones finales

Para el desarrollo de este resultado se empleó la funcionalidad *StopAnalyzer* de Lucene, el cual fue configurado con el diccionario que usa por defecto el *SpanishAnalyzer* de Lucene. Dicho diccionario contiene 330 palabras entre artículos, preposiciones, conjunciones, pronombres, y conjugaciones de algunos verbos básicos (ser, estar, haber, tener, entre otros) del idioma español. Si bien este diccionario es editable, para efectos de este entregable no se ha ahondado en la modificación de las palabras incluidas por defecto en él.

Sin embargo, cabe resaltar que existen ciertos escenarios en los que tras un análisis más profundo se puede concluir apropiado excluir del diccionario algunas de las palabras que en un principio sí fueron consideradas *stopwords*. Por ejemplo, tal es el caso de la palabra “no” en la consulta “no programación en pascal”. Si bien la palabra “no” puede ser considerada un *stopword*, su presencia para mantener la integridad semántica de la consulta completa sí se considera un factor relevante a tener en cuenta. No obstante, para efectos de este proyecto dichos escenarios particulares no tendrán impacto sobre los resultados, ya que como se mencionó en la sección 3 del capítulo 1, el análisis de la consulta por lógica proposicional no está incluido en el

alcance de este proyecto.

Finalmente, cabe resaltar que si bien para este proyecto se excluyó del alcance el análisis exhaustivo de las palabras a incluir en el diccionario de *stopwords* con la consideración de sus implicancias, se propone su desarrollo en futuros trabajos.

1.2 Resultado esperado 2: Mecanismo de pre-procesamiento de la consulta en lenguaje natural que reduzca las palabras a su forma base o lema (proceso de lematización).

Este mecanismo de procesamiento del lenguaje natural permitirá simplificar tanto la consulta del usuario como las frases clave de las fuentes de información, de tal forma que la diversidad de representaciones de un mismo concepto derivadas de los principios sintácticos del lenguaje español puedan ser reducidas a una forma base o diccionario (*lemma*), y que por tanto, pueda encontrarse la equivalencia entre las palabras de la consulta y las de las fuentes con independencia de su representación original.

1.2.1 Pre-requisitos

Este mecanismo emplea como input la consulta pre-procesada sin stopwords.

1.2.2 Funcionamiento

En primer lugar, cada *token* de la consulta sin *stopwords* es analizada con ayuda de un componente que recupera el *lemma* respectivo de un diccionario de lemas en español. Si efectivamente se encuentra un *lemma* asociado, se reemplaza el *token* original por el *lemma*. Si no se encontró, el *token* original se mantiene en la consulta.
Finalmente, este mecanismo retorna la consulta pre-procesada, es decir, la consulta del usuario sin incluir los *stopwords* en español removidos por el mecanismo anterior y con los *token* convertidos a su forma base.

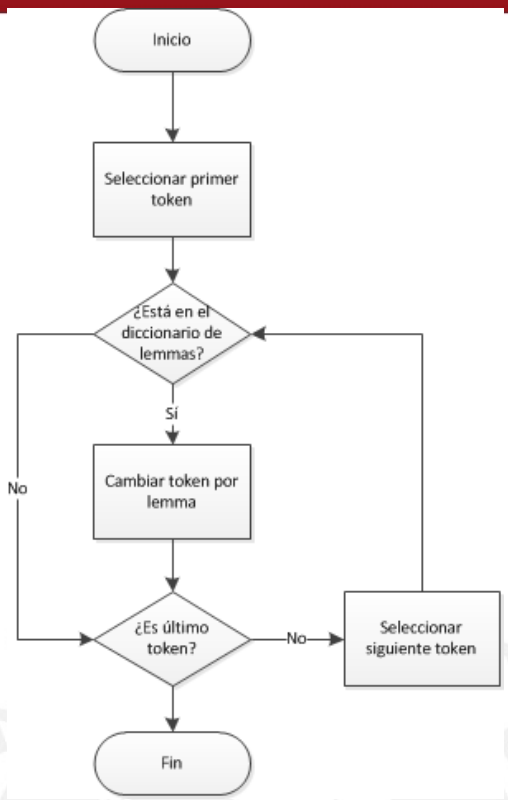


Imagen 3. Mecanismo de lematización

1.2.3 Mecanismo de prueba

Para la verificación del mecanismo se realizó la prueba con la siguiente consulta sin *stopwords*:

Input (Consulta sin *stopwords*):

Consulta sin stopwords: objetivos resultado eliminar palabras conocidas stopwords relevantes proceso recuperación

Output (Consulta final):

Consulta lematizada: objetivo resultado eliminar palabra conocer stopwords relevante procesar recuperación

De estos resultados se puede observar que las palabras lematizadas fueron: “objetivos” por “objetivo”

“palabras” por “palabra”
“conocidas” por “conocer”
“relevantes” por “relevante”
“proceso” por “procesar”

De los resultados obtenidos se puede observar que las palabras fueron lematizadas correctamente, ya que las palabras resultantes se estandarizaron a su forma singular y base.

1.2.4 Consideraciones finales

Para el desarrollo de este resultado se empleó el diccionario de *lemmas* en español por defecto de Freeling, y en particular la función *analyze*. Cabe mencionar que este diccionario en español contiene el mapa de conjugaciones del español de España, más no el latinoamericano, por lo que las consultas sobre las cuales se puede obtener los *lemmas* está sujeto a esta restricción.

1.3 Conclusiones

La consulta pre-procesada obtenida luego de la lematización, ya que se encuentra estandarizada y contiene sólo palabras relevantes para la búsqueda, es una correcta base sobre la cual continuar con el proceso de expansión.

CAPÍTULO 4

1 Objetivo específico 2: Diseñar un mecanismo que permita resolver la ambigüedad de las palabras en un dominio en específico.

Este objetivo está orientado a atacar otra de las causas identificadas como raíz de la problemática, la cual es la dificultad de los sistemas de recuperación información para lidiar con una de las características del lenguaje natural: ambigüedad de las palabras (homonimia).

Un caso concreto de ambigüedad se puede identificar en la homonimia, y a manera de ejemplificación, al plantear en una consulta el término “arreglo”. Esta palabra puede hacer referencia al concepto de la acción de “arreglar”, al concepto de coordinación o conciliación, o a una estructura de datos en el ámbito de la informática, entre otros. Más aún, dentro del mismo ámbito de la informática, al mencionarse el término “arreglo”, éste puede hacer referencia a distintos tipos de estructuras de datos con características particulares como los arreglos unidimensionales, multidimensionales o los arreglos de múltiples subíndices.

Lo que se espera con el desarrollo de este objetivo es poder identificar el concepto al que está asociada cierta palabra de la consulta del usuario, y asimismo poder brindar un primer nivel de evolución de la consulta pre-procesada a la consulta expandida.

1.1 Resultado esperado 1: Modelo de desambiguación de palabras en lenguaje natural dentro de un dominio usando ontologías.

El modelo de desambiguación planteado permitirá, con cierto grado de certeza, identificar el concepto al que hace referencia uno o varios términos de la consulta del usuario, términos los cuales pueden considerarse ambiguos dentro de un dominio de conocimiento debido a la característica de homonimia propia del lenguaje natural ejemplificada en el punto anterior.

Para este modelo, se ha decidido basar la desambiguación en 2 componentes de soporte:
En primer lugar, en el uso de los términos complementarios (términos denominados no-ambiguos respecto al término ambiguo analizado) de la consulta y, en segundo lugar, empleando la característica de sinonimia del lenguaje natural a favor de la

desambiguación. Las características de la ontología empleadas son las relaciones entre el concepto (nodo) ambiguo y complementario de la ontología, y la propiedad de equivalencia entre conceptos de la ontología.

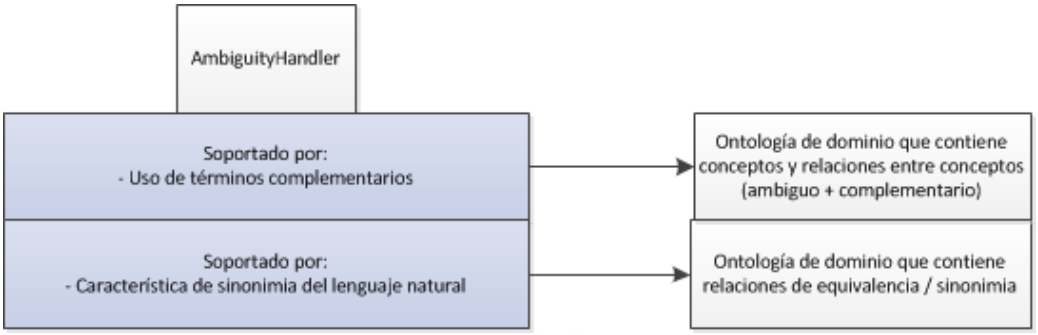


Imagen 4. Componentes de soporte y la respectiva característica empleada de la ontología.

Para el primer caso, se utilizará una ontología de dominio sobre la cual se identificará los posibles conceptos a los que puede estar haciendo referencia el término ambiguo en análisis, y con apoyo de los términos complementarios se establecerá un ranking donde el término con mejor puntaje representará el concepto más apropiado para la palabra ambigua.

Asimismo, para efectos de la desambiguación, el cálculo del puntaje de similitud respecto a cada posible concepto del término ambiguo se apoyará en la asociación entre las palabras complementarias de la consulta y los sinónimos de dichos posibles conceptos.

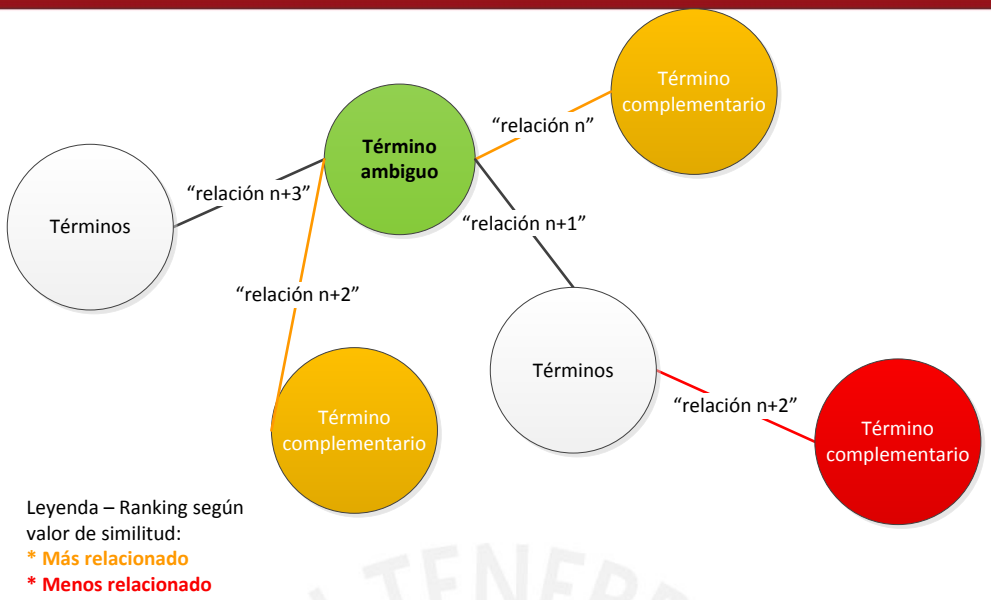


Imagen 5. Nodos y relaciones de la ontología con representación de semáforo según ranking resultante.

1.1.1 Pre-requisitos

Este mecanismo emplea como input:

La palabra ambigua de la consulta del usuario.

Una o más palabras complementarias de la consulta del usuario.

La ontología de dominio.

1.1.2 Funcionamiento

En la imagen 6 se puede observar el flujo general del mecanismo de desambiguación planteado, el cual fue realizado a partir del flujo general de la propuesta de Zhao Lu para la desambiguación de nombres personales chinos. [LU, et al 2013]

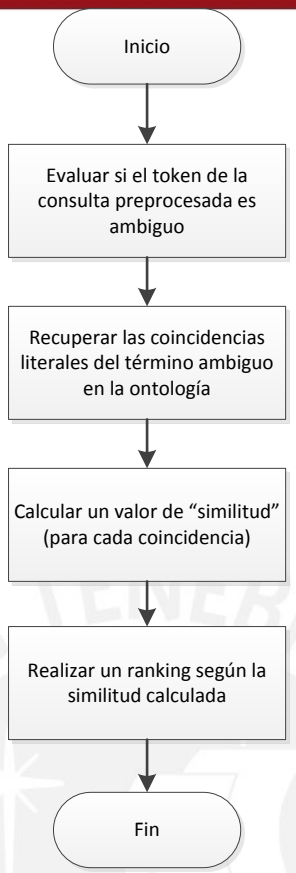


Imagen 6. Representación general del mecanismo de desambiguación.

En primer lugar, cada palabra de la consulta pre-procesada pasa por una primera evaluación que verifica si dicho término es ambiguo, considerándolo como ambiguo si se encuentra más de 1 vez en nodos diferentes de la ontología de dominio.
Si es que dicho término se encuentra a lo mucho 1 vez en la ontología, no es necesario que dicha palabra pase por el proceso de desambiguación, caso contrario, se procede a calcular la similitud acumulada de cada una de esas coincidencias retornadas.

El cálculo de la similitud se realiza con apoyo de los términos complementarios de la consulta. Dicho cálculo consiste en realizar una evaluación recursiva de los nodos de la ontología, y en cada nivel evaluar si la palabra complementaria se encuentra en el nodo analizado o en sus términos equivalentes, caso contrario se procede a analizar el respectivo nodo padre siguiendo el mismo criterio. Para los casos en los que la coincidencia analizada contiene también la palabra complementaria, la similitud es máxima (definida como 0). Por otro lado, si la coincidencia no contiene la palabra complementaria, se procede a buscar en los nodos jerárquicamente superiores y que

tienen una relación con el elemento evaluado. En esos casos la similitud se define como = CONSTANTE_REC * nivel, donde nivel es el nivel de recursividad en el que se encontró la coincidencia, de tal forma que los niveles más alejados sean calificados como menos similares, con lo que finalmente el cálculo de la similitud (y por ende la recursividad) termina.
Otra condición de parada definida se considera al llegar a un nivel máximo (tope) de recursividad sin haberse encontrado alguna coincidencia de la palabra complementaria en ninguno de los niveles.

Finalmente, se procede a realizar un ranking entre los valores de similitud calculados, donde el primer elemento del ranking (cuya similitud sea más cercano a 0) será el concepto más cercano / acertado.

Este mecanismo, siguiendo la técnica de expansión de consultas, añadirá dicho nuevo concepto a la que se conocerá como consulta expandida.

Para el desarrollo de este resultado se empleó el lenguaje de consultas SPARQL, tanto para la recuperación de las coincidencias del término ambiguo en la ontología, como para la identificación del siguiente nodo padre (relacionado al nodo analizado) en el proceso recursivo para el cálculo de la similitud.

1.1.3 Mecanismo de prueba

- Como parte de la configuración del mecanismo de prueba se definió como tope de la recursividad 5 niveles.
- Se estableció el valor de 3 a la CONSTANTE_REC empleada en el cálculo de la similitud. El objetivo de la existencia de esta constante será descrita en la sección 1.2.1 del capítulo 6.
- Se empleó la ontología descrita en la sección 1.1.1 del capítulo 5.

Para la verificación del mecanismo se realizaron las siguientes pruebas:

Caso A

Prueba A1 – Sin expansión

Input:

Consulta inicial:	pilas
Consulta lematizada:	**pila**

Output:

Incluir conocimiento del usuario activado: No
Palabra ambigua: pila
Palabra complementaria: -
***** RANKING *****
ID: AplicacionesPilas Nombre: Pilas **Lemma: pila** **Valor: 0.0** Nivel: 1
ID: TADPilas Nombre: Pilas **Lemma: pila** **Valor: 0.0** Nivel: 1

Query expandido: pila

Prueba A2 – Con expansión por término complementario

Input:

Consulta inicial: aplicaciones pilas
Consulta lematizada: **aplicación pila**

Output:

Incluir conocimiento del usuario activado: No
Palabra ambigua: pila
Palabra complementaria: aplicación

***** RANKING *****
ID: AplicacionesPilas **Lemma: aplicación pila** **Valor: 2.0**
ID: TADPilas Lemma: tipo abstracto pila Valor: 16.0
***** RANKING *****
ID: AplicacionesPilas **Lemma: aplicación pila** **Valor: 2.0**
ID: AplicacionesArbolesBinariosBusquedaGrafos
Lemma: aplicación arbolar binario busqueda grafo Valor: 16.0
ID: AplicacionEstructurasDatos Lemma: aplicación estructura dato Valor: 16.0
ID: AplicacionesArboles Lemma: aplicación arbolar Valor: 16.0
ID: AplicacionesListas Lemma: aplicación lista Valor: 16.0
ID: AplicacionesColas Lemma: aplicación cola Valor: 16.0
ID: AplicacionesRecorridosMetodosEspecialesArbolesGrafos
Lemma: aplicación recorrer metodos busqueda especial arbolar grafo Valor: 16.0

Query expandido: pila aplicación aplicación pila

Caso B

Prueba B1

Input:

Consulta inicial:	archivos
Consulta lematizada:	**archivo**

Output:

Incluir conocimiento del usuario activado: No

Palabra ambigua: archivo

Palabra complementaria: -

***** RANKING *****

ID: ArchivosTexto	Nombre: Archivos de texto	**Lemma: archivo texto**	**Valor: 0.0**
ID: OrdenacionArchivos	Nombre: Ordenacion de archivos	**Lemma: Ordenacion archivo**	**Valor: 0.0**
ID: Archivos	Nombre: Archivos	**Lemma: archivo**	**Valor: 0.0**
ID: ArchivosDescriptor	Nombre: Descriptores de archivos	**Lemma: descriptor archivo**	**Valor: 0.0**
ID: ArchivosBinarios	Nombre: Archivos binarios	**Lemma: archivo binario**	**Valor: 0.0**

Query expandido: archivo texto archivo descriptor archivo ordenacion archivo archivo binario

Prueba B2

Input:

Consulta inicial:	archivos algoritmia
Consulta lematizada:	**archivo algoritmia**

Output:

Incluir conocimiento del usuario activado: No

Palabra ambigua: archivo

Palabra complementaria: algoritmia

***** RANKING *****

ID: OrdenacionArchivos	**Nombre: Ordenacion de archivos**	**Lemma: Ordenacion archivo**	**Valor: 10.0**
ID: ArchivosTexto	Nombre: Archivos de texto	Lemma: archivo texto	Valor: 16.0
ID: Archivos	Nombre: Archivos	Lemma: archivo	Valor: 16.0
ID: ArchivosDescriptor	Nombre: Descriptores de archivos	Lemma: descriptor archivo	Valor: 16.0
ID: ArchivosBinarios	Nombre: Archivos binarios	Lemma: archivo binario	Valor: 16.0

Query expandido: algoritmia archivo ordenacion archivo

Antes de iniciar la interpretación y evaluación de los resultados cabe mencionar que el ID mostrado en los recuadros de *Output* corresponde al código identificador de los nodos dentro de la ontología. Este campo no es utilizado dentro de la lógica de ninguno de los mecanismos propuestos en este proyecto, pero se muestra para que pueda reconocerse sin dificultad que los nodos mostrados corresponden a diferentes conceptos, a pesar de que sus nombres sean iguales entre sí (reflejo de la característica de homonimia del lenguaje natural).

En los resultados de las pruebas A1 y B1 se puede observar que el mecanismo efectivamente halló en la ontología todas las coincidencias de los términos ambiguos “pilas” y “archivos”, respectivamente. Sin embargo, dichas consultas no brindan mayor información que contribuya a poder identificar entre todas las opciones, el término más apropiado. Los resultados muestran que en cada caso, todos los conceptos obtuvieron un mismo valor de similitud igual a 0, dado que al no contar con una palabra complementaria, ya no se siguió recorriendo a través de los distintos niveles que van incrementando el valor de la similitud.

Mientras tanto, en los resultados de la prueba A2 se puede observar que se logró identificar como término más apropiado al que hacía referencia a la palabra ambigua en un nivel y la complementaria. Los otros conceptos tuvieron el nivel de similitud más alejado posible, ya que en la evaluación de ninguno de sus nodos relacionados, en ningún nivel tuvo relación alguna con el término complementario empleado para la desambiguación.

De forma análoga, en los resultados de la prueba B2 se puede observar que se logró identificar como término más apropiado al que hacía referencia a la palabra ambigua en un nivel, y a la complementaria en un nivel superior, siendo una relación derivada por transitividad la que une ambos nodos (ID:Algoritmia “tieneProgramaAnalitico” ID:INF263 “tieneUnidadAprendizaje” ID:Ordenacion “tieneConcepto” ID: OrdenacionArchivos).

Tanto en las pruebas A2 como B2 los conceptos no relacionados obtuvieron un valor de similitud igual a 16 (CONSTANTE_REC*cantidad_niveles + 1). Esto ya que se recorrió una cantidad de niveles hasta llegar al tope (predefinido como 5) sin encontrar

alguna coincidencia de la palabra complementaria en alguno de ellos, asimismo, el 1 final siempre se adiciona debido a la relación intrínseca entre la palabra de ambigua y su coincidencia literal consigo misma.

1.2 Conclusiones

Por el análisis de los resultados según lo expuesto se ha podido observar que los resultados cumplen con el objetivo específico planteado, ya que dadas las condiciones limitadas por la consulta se ha podido encontrar un concepto al que puede hacer referencia la consulta, con lo cual se obtienen términos para mejorar la consulta a través de la expansión y se descartan conceptos que podrían generar ruido innecesario si fueran añadidos como términos a la consulta expandida.

CAPÍTULO 5

1 Objetivo específico 3: Diseñar un mecanismo que permita al usuario estructurar consultas sin la necesidad de comprender el dominio completo del conocimiento y el dominio del motor de búsqueda.

Este objetivo está orientado a atacar otra de las causas identificadas como raíz de la problemática, la cual es que el usuario no siempre logra traducir sus palabras en una consulta 'adecuada', debido a la dificultad para estructurar sus ideas y a que los usuarios no cuentan con el conocimiento completo del dominio sobre el cual tienen dudas o necesidades de información.

Un primer elemento que permite superar la dificultad del usuario para traducir sus palabras en una consulta adecuada y adaptada a lo que espera el motor de búsqueda es facilitado por los mecanismos planteados para el objetivo 1, gracias al cual el usuario tiene la libertad de poder redactar su consulta en lenguaje natural y más bien es el mecanismo de búsqueda quien busca adaptarse y poder lidiar con la consulta en lenguaje natural del usuario. Sin embargo, aún con estos mecanismos subsiste el hecho de que el usuario no cuenta con el conocimiento completo del dominio, lo cual representa una dificultad para un proceso de recuperación efectivo.

Por esta razón, lo que se espera con el desarrollo de este objetivo es poder establecer la base que permita el posterior procesamiento del conocimiento del dominio (conocimiento con el que el usuario no cuenta) pero que sí se encuentra representado en una estructura en la forma de ontología de dominio.

1.1 Resultado esperado 1: Estructura para almacenar el conocimiento del dominio.

Este resultado está compuesto por dos elementos. El primero, la ontología que contiene la información del dominio, y el segundo, el conjunto de estructuras necesarias para soportar la recuperación y almacenamiento temporal de la información a ser extraída en distintos momentos del proceso de preparación para la expansión.

1.1.1 Estructura propuesta:

a) Ontología de dominio:

Para fines del desarrollo de este y otros dos proyectos de fin de carrera, se creó una ontología en el dominio de los conceptos estudiados en la rama de ciencias de la computación de la especialidad de ingeniería informática de la Pontificia Universidad Católica del Perú, en formato OWL/RDF. Esta ontología cuenta con:

- Una propiedad "NombrePreferente" que contiene el nombre principal de cada concepto. Cada nodo de la ontología tiene exactamente un nombre preferente asociado a él. Este elemento existe con la única finalidad de ser mostrado al usuario final para su fácil lectura y comprensión del concepto al que se quiere hacer referencia. Sin embargo, para realizar todas las búsquedas y recorridos sobre la ontología se utilizarán solamente los valores asociados a las propiedades "lemma" y "sinónimos", las cuales contienen la versión lematizada y sin *stopwords* del nombre preferente, y de las equivalencias del nombre preferente, respectivamente.
- Una propiedad "Sinonimos", que relaciona cada nodo con sus respectivos términos equivalentes en versiones lematizadas y sin *stopwords*. Cada nodo de la ontología puede tener uno o varios sinónimos asociados a él.
- Una propiedad "Lemma", la cual relaciona cada concepto con su respectiva denominación en forma base (*lemma*). Cada nodo de la ontología tiene exactamente un *lemma* asociado a él.

Si bien el valor asociado a la propiedad nombre preferente de cada nodo es completado en base al conocimiento especializado propio de quien diseña la ontología, las formas lematizadas a colocar en las otras dos propiedades se generan empleando los mecanismos propuestos en las secciones 1.1.1 y 1.1.2 del capítulo 3. Luego de obtener dichos *lemmas,* estos deben ser introducidos manualmente en la ontología como parte del proceso de creación de la ontología.

La razón por la que se optó por incluir la versión lematizada de los conceptos dentro de la misma ontología es porque el tiempo de ejecución del mecanismo de lematización sumado al de remoción de *stopwords,* cuando son aplicados sobre los conceptos de una estructura como la de la ontología creada (con alrededor de 70 nodos), es considerable (tiempo > 10 segundos), y más aún, se eleva de forma significativa mientras más comparaciones se deba hacer a causa de una mayor cantidad de palabras en la consulta inicial.

A continuación se muestran los diagramas y tablas que detallan la estructura de objetos y propiedades de la ontología:

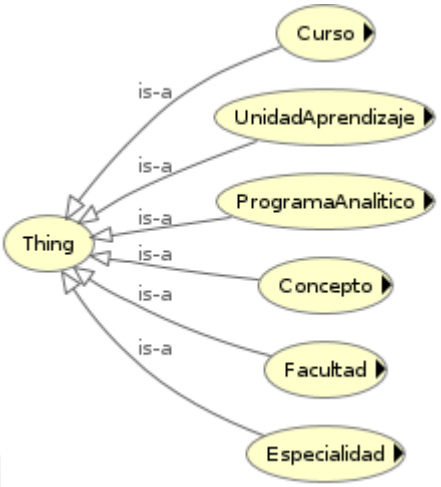
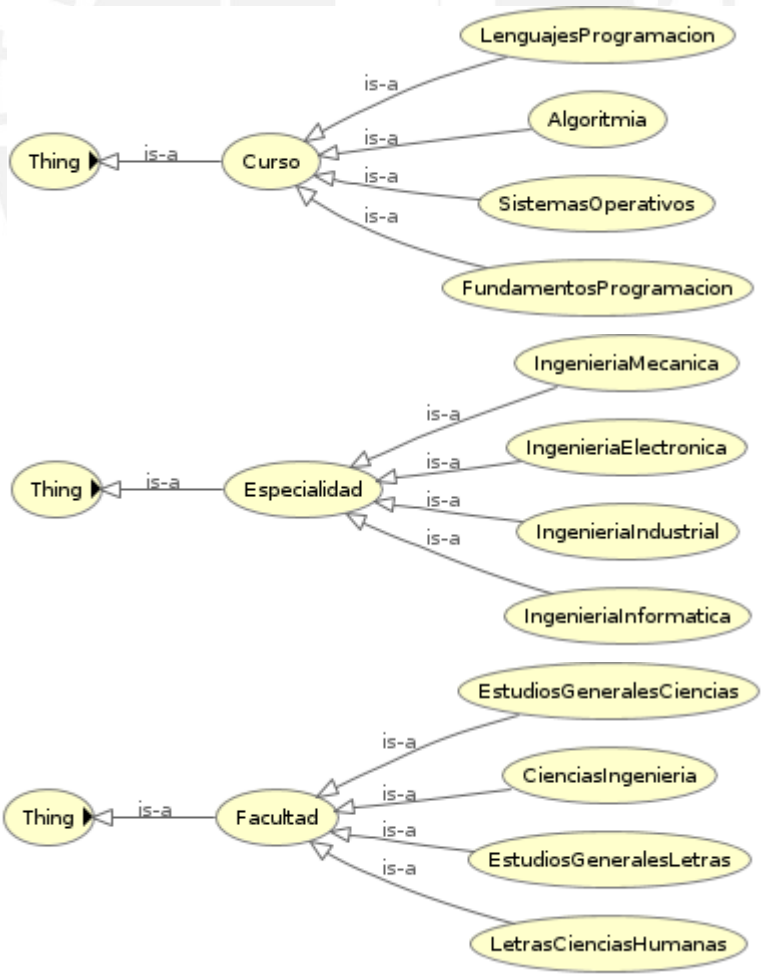


Imagen 7. Estructura global de las clases principales de la ontología



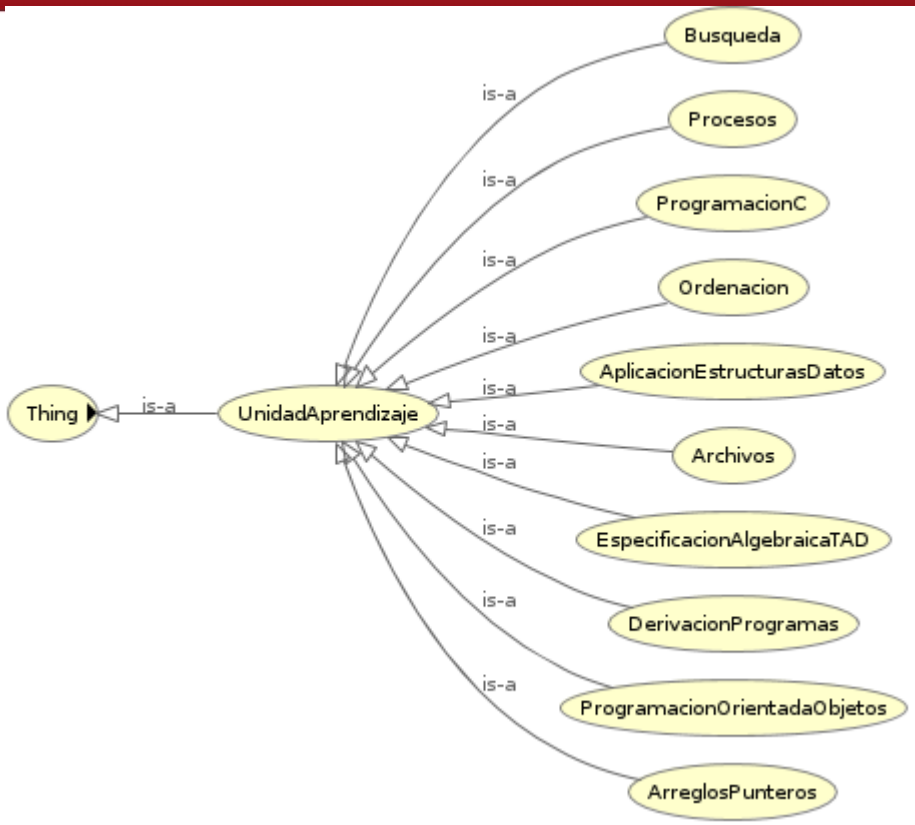


Imagen 8. Estructura de subclases de la ontología

instanciadominio	instanciarango
pucp:CienciasIngenieria	pucp:IngenieriaMecanica
pucp:CienciasIngenieria	pucp:IngenieriaInformatica
pucp:CienciasIngenieria	pucp:IngenieriaIndustrial
pucp:CienciasIngenieria	pucp:IngenieriaElectronica

Tabla 6. Nodos relacionados por propiedad tieneEspecialidad

instanciadominio	instanciarango
pucp:IngenieriaInformatica	pucp:FundamentosProgramacion
pucp:IngenieriaInformatica	pucp:SistemasOperativos
pucp:IngenieriaInformatica	pucp:Algoritmia
pucp:IngenieriaInformatica	pucp:LenguajesProgramacion

Tabla 7. Nodos relacionados por propiedad tieneCurso

instanciadominio	instanciarango
pucp:LenguajesProgramacion	pucp:INF281
pucp:SistemasOperativos	pucp:INF239
pucp:FundamentosProgramacion	pucp:INF220
pucp:Algoritmia	pucp:INF263

Tabla 8. Nodos relacionados por propiedad tieneProgramaAnalitico

instanciadominio	instanciarango
pucp:INF239	pucp:Procesos
pucp:INF281	pucp:ProgramacionOrientadaObjetos
pucp:INF281	pucp:ArreglosPunteros
pucp:INF281	pucp:Archivos
pucp:INF220	pucp:DerivacionProgramas
pucp:INF220	pucp:EspecificacionAlgebraicaTAD
pucp:INF263	pucp:Busqueda
pucp:INF263	pucp:ProgramacionC
pucp:INF263	pucp:AplicacionEstructurasDatos
pucp:INF263	pucp:Ordenacion

Tabla 9. Nodos relacionados por propiedad tieneUnidadDeAprendizaje

instanciadominio	instanciarango
pucp:AplicacionEstructurasDatos	pucp:AplicacionesRecorridosMetodosEspecialesArbolesGrafos
pucp:AplicacionEstructurasDatos	pucp:AplicacionesColas
pucp:AplicacionEstructurasDatos	pucp:AplicacionesListas
pucp:AplicacionEstructurasDatos	pucp:AplicacionesArbolesBinariosBusquedaGrafos
pucp:AplicacionEstructurasDatos	pucp:AplicacionesPilas
pucp:ArreglosPunteros	pucp:AYPCadenasCaracteres
pucp:ArreglosPunteros	pucp:AYPPunterosFunciones
pucp:ArreglosPunteros	pucp:AYPPunterosGenericos
pucp:DerivacionProgramas	pucp:DerivacionLenguajeOrdenesConGuarda
pucp:DerivacionProgramas	pucp:DerivacionCalculoPredicados
pucp:Busqueda	pucp:BusquedaSecuencial
pucp:Busqueda	pucp:BusquedaTransformacionLlaves

pucp:Busqueda	pucp:BusquedaTratamientoColisiones
pucp:Busqueda	pucp:BusquedaBinaria
pucp:Busqueda	pucp:BusquedaFuncionesHash
pucp:Busqueda	pucp:BusquedaTablasHash
pucp:Archivos	pucp:ArchivosBinarios
pucp:Archivos	pucp:ArchivosTexto
pucp:Archivos	pucp:ArchivosDescriptor
pucp:Procesos	pucp:ProcesosSemaforos
pucp:Procesos	pucp:ProcesosExclusionMutua
pucp:Procesos	pucp:ProcesosMonitor
pucp:EspecificacionAlgebraicaTAD	pucp:TADConjuntos
pucp:EspecificacionAlgebraicaTAD	pucp:TADListas
pucp:EspecificacionAlgebraicaTAD	pucp:TADGrafos
pucp:EspecificacionAlgebraicaTAD	pucp:TADColas
pucp:EspecificacionAlgebraicaTAD	pucp:TADPilas
pucp:EspecificacionAlgebraicaTAD	pucp:TADVectores
pucp:ProgramacionOrientadaObjetos	pucp:POOSobrecarga
pucp:ProgramacionOrientadaObjetos	pucp:POOAtributos
pucp:ProgramacionOrientadaObjetos	pucp:POOClases
pucp:ProgramacionOrientadaObjetos	pucp:POOMetodos
pucp:Ordenacion	pucp:OrdenacionSeleccion
pucp:Ordenacion	pucp:OrdenacionArchivos
pucp:Ordenacion	pucp:OrdenacionInsercion
pucp:Ordenacion	pucp:OrdenacionMonticulos
pucp:Ordenacion	pucp:OrdenacionIntercambio

Tabla 10. Nodos relacionados por propiedad tieneConcepto

Asimismo, en las tablas 11 y 12 se puede observar el mapa de *sinónimos* y *lemmas* por cada clase o subclase de la ontología:

Clase	Lemma
pucp:AYPPunterosFunciones	"puntero funcionar"
pucp:TADListas	"lista"
pucp:POOAtributos	"atributo"
pucp:BusquedaFuncionesHash	"Busqueda funcionar hash"
pucp:AYPCadenasCaracteres	"cadena carácter"
pucp:Algoritmia	"algoritmia"
pucp:TADVectores	"vector"
pucp:SistemasOperativos	"sistema operativo"
pucp:INF239	"INF239"
pucp:ArchivosTexto	"archivo texto"
pucp:OrdenacionMonticulos	"Ordenacion monticulos"
pucp:OrdenacionInsercion	"Ordenacion insercion"
pucp:BusquedaTratamientoColisiones	"tratamiento colisionar"
pucp:OrdenacionSeleccion	"Ordenacion seleccion"
pucp:BusquedaTablasHash	"tabla Hash"
pucp:ProgramacionC	"programacion c"
pucp:AplicacionesArbolesBinariosBusquedaGrafo	"arbolar binario busqueda grafo"
pucp:ProgramacionOrientadaObjetos	"Programacion orientar objeto"
pucp:POOMetodos	"Metodos"
pucp:LenguajesProgramacion	"lenguaje Programacion 1"
pucp:ArchivosDescriptor	"descriptor archivo"
pucp:ProcesosMonitor	"monitor"
pucp:ProgramaAnalitico	"programa analitico"
pucp:OrdenacionArchivos	"Ordenacion archivo"
pucp:Busqueda	"Busqueda"
pucp:Concepto	"concepto"
pucp:DerivacionCalculoPredicados	"calcular predicado"
pucp:OrdenamientoRapido	"ordenamiento rapido"
pucp:INF220	"INF220"
pucp:UnidadAprendizaje	"unidad aprendizaje"

pucp:BusquedaTransformacionLlaves	"Transformacion llave"
pucp:ProcesosExclusionMutua	"Exclusion mutuo"
pucp:OrdenacionIntercambio	"Ordenacion intercambiar"
pucp:DerivacionLenguajeOrdenesConGuarda	"lenguaje ordenar guarda"
pucp:LetrasCienciasHumanas	"letra ciencia humanar"
pucp:Archivos	"archivo"
pucp:Ordenacion	"Ordenacion"
pucp:AplicacionesPilas	"pila"
pucp:AplicacionEstructurasDatos	"aplicación estructura dato"
pucp:INF263	"INF263"
pucp:ArreglosPunteros	"arreglo puntero"
pucp:ProcesosSemaforos	"Semaforos"
pucp:TADConjuntos	"conjunto"
pucp:Facultad	"facultad"
pucp:FundamentosProgramacion	"fundamento Programacion"
pucp:Especialidad	"especialidad"
pucp:EstudiosGeneralesCiencias	"estudio general ciencia"
pucp:AplicacionesListas	"lista"
pucp:TADPilas	"pila"
pucp:POOClases	"clase"
pucp:POOSobrecarga	"sobrecarga"
pucp:IngenieriaMecanica	"Ingenieria mecanica"
pucp:AplicacionesColas	"cola"
pucp:AplicacionesRecorridosMetodosEspecialesArbolesGrafos	"recorrer metodos busqueda especial arbolar grafo"
pucp:AplicacionesArboles	"arbolar"
pucp:CienciasIngenieria	"ciencia Ingenieria"
pucp:IngenieriaInformatica	"Ingenieria Informatica"
pucp:EstudiosGeneralesLetras	"estudio general letra"
pucp:BusquedaSecuencial	"Busqueda secuencial"
pucp:AYPPunterosGenericos	"puntero género"
pucp:IngenieriaElectronica	"Ingenieria Electronica"
pucp:ArchivosBinarios	"archivo binario"
pucp:Curso	"cursar"
pucp:INF281	"INF281"
pucp:IngenieriaIndustrial	"Ingenieria industrial"
pucp:DerivacionProgramas	"Derivacion programa"
pucp:TADColas	"cola"
pucp:Procesos	"proceso"
pucp:EspecificacionAlgebraicaTAD	"Especificacion algebraico tipo dato abstracto"
pucp:TADGrafos	"grafo"
pucp:BusquedaBinaria	"Busqueda binario"

Tabla 11. Mapa de *lemma* por nodo

class	sinonimos
pucp:AYPCadenasCaracteres	"strings"
pucp:Algoritmia	"Algoritmo"
pucp:SistemasOperativos	"Operativo"
pucp:ProgramacionOrientadaObjetos	"OOP"
pucp:ProgramacionOrientadaObjetos	"POO"
pucp:LenguajesProgramacion	"LP1"
pucp:OrdenamientoRapido	"quicksort"
pucp:ProcesosExclusionMutua	"Mutex"
pucp:DerivacionLenguajeOrdenesConGuarda	"GCL"
pucp:Archivos	"Fichero"
pucp:Ordenacion	"Ordenamiento"
pucp:ArreglosPunteros	"Pointer"
pucp:ArreglosPunteros	"Array"
pucp:FundamentosProgramacion	"fp"
pucp:FundamentosProgramacion	"Fundamento"
pucp:FundamentosProgramacion	"Funda"

pucp:EstudiosGeneralesCiencias	"EEGGCC"
pucp:EstudiosGeneralesLetras	"EEGGLL"
pucp:EspecificacionAlgebraicaTAD	"Especificacion algebraico tipo abstracto dato"
pucp:EspecificacionAlgebraicaTAD	"Especificacion algebraico TAD"

Tabla 12. Mapa de sinónimos por nodo

b) Estructura para soportar la recuperación y almacenamiento temporal de la información:

La estructura propuesta, y cuyo uso será explicado en resultados posteriores se muestra a continuación:

```
NodoBase {
        String stURI;
        String stNombrePreferente;
        String stLemma;
}

Clase {
        NodoBase nodo;
        List<NodoBase> lstSinonimos;
        Padre nodoPadre;
        double valorSimilitud;
        Integer nivel = 1;
        encontrado = false;
        acumuladoSimilitud = 0;
        List<Concepto> lstDesambiguados;
}

Padre {
        NodoBase nodoDominio;
        NodoBase nodoInstanciaDominio;
        NodoBase nodoPropiedad;
        NodoBase nodoRango;
}

Concepto {
        NodoBase nodo;
```

```
        List<NodoBase> lstDesambiguados;
        List<NodoBase> lstSinonimos;
}

ResultadoExpansion {
        List<Concepto> lstConceptos;
        List<String> expansionEquivalencias;
        List<String> expansionNiveles;
        List<String> expansionConocimientoUsuario;

        List<Concepto> lstQueryExpandidoPlano;
}
```

1.2 Resultado esperado 2: Modelo que permita recuperar el conocimiento del dominio a partir de una estructura en la que se encuentra codificada.

La recuperación del conocimiento a partir de la estructura en la que se encuentra codificada (ontología) se da a través de dos interacciones, siendo el primer momento al recuperar de la ontología la lista de elementos coincidentes (o no coincidentes) respecto a la palabra ambigua según el procedimiento descrito para el mecanismo de desambiguación y de identificación de equivalencias. El segundo momento en el que se tiene interacción con la estructura ontológica es al momento de recuperar la instancia padre de un nodo también según el mecanismo recursivo de desambiguación.

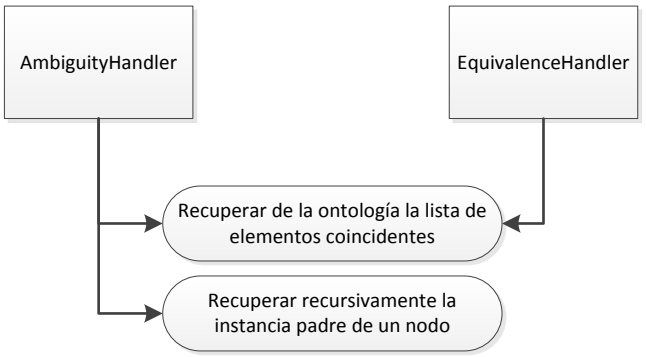


Imagen 9. Momentos de interacción de cada componente con la ontología.

1.2.1 Funcionamiento

Para el caso del primer momento de interacción, dicha recuperación se realiza haciendo una consulta a la ontología de tal forma que se recupere tanto el nombre "Nombre Preferente" de cada nodo de la ontología como sus respectivos sinónimos. A continuación, se realiza un recorrido de los resultados obtenidos y se aplica a dicho nombre preferente y a los sinónimos el mecanismo de remoción de *stopwords* y lematización. Este resultado es almacenado en una estructura de tipo "Clase", guardando en un NodoBase el nombre preferente y el URI del elemento para futuras referencias, y la lista de sinónimos de dicho concepto en una lista de tipo NodoBase denominada sinónimos.

Para el caso del segundo momento de interacción, dicha recuperación se realiza haciendo una consulta a la ontología de tal forma que se recupere todos los nodos que estén asociados al nodo analizado a través de una propiedad cualquiera existente en la ontología. Cada elemento recuperado se almacena en una estructura Padre cuyo elemento principal para el mecanismo de desambiguación es el elemento nodoInstanciaDominio de tipo NodoBase. Es decir, es en este atributo que se almacena el nodo padre hallado.

1.2.2 Mecanismo de prueba

Este modelo ha sido probado a través de su uso en el mecanismo de desambiguación anteriormente expuesto.

1.3 Conclusiones

Este modelo se ha probado apropiado ya que ha facilitado el lograr completar con éxito el proceso de desambiguación, el cual está basado en la información recuperada con éxito de la ontología y almacenada en las estructuras propuestas en los resultados para el presente objetivo.

CAPÍTULO 6

1 Objetivo específico 4: Diseñar un mecanismo que permita al sistema IR comprender las necesidades de información del usuario.

Este objetivo busca superar una de las causas identificadas en la problemática, la cual es la dificultad para interpretar lo que el usuario quiere y necesita realmente, ya sea por desconocimiento del contexto de la búsqueda o por desconocimiento del usuario.

Lo que se espera con el desarrollo de este objetivo es poder adicionar a la consulta (como parte de la expansión) un conjunto apropiado de términos que pueda conllevar a una mejor interpretación de lo que el usuario quiere y necesita realmente al provenir estos términos de una fuente confiable como lo es el conocimiento del mismo usuario.

1.1 Resultado esperado 1: Estructura para almacenar el conocimiento previo del usuario.

La fuente que contiene la información del usuario se ha definido como una base de datos relacional, de tal forma que simule la realidad de un sistema que gestiona la matrícula de los alumnos de una universidad.

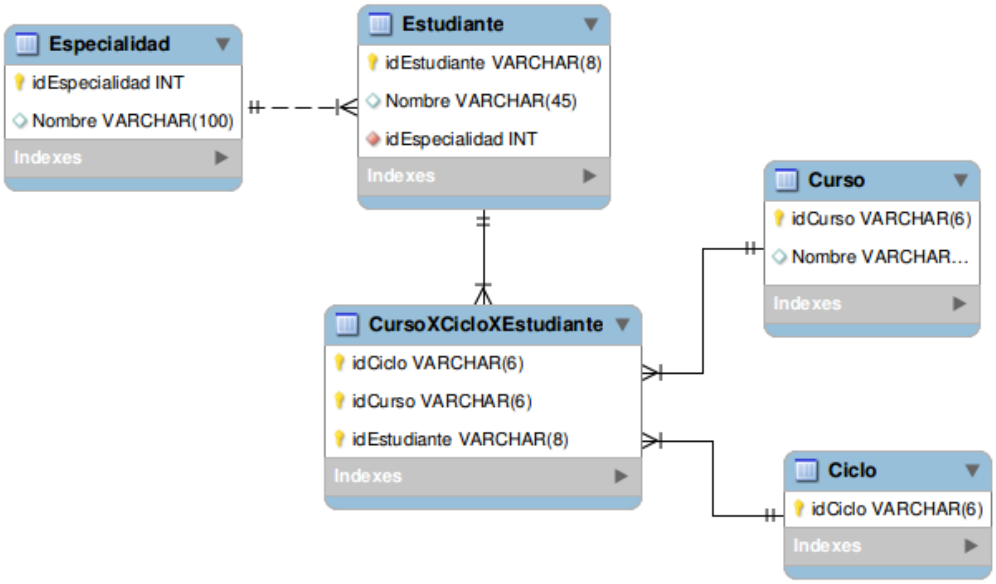


Imagen 10. Diagrama entidad relación de una base de datos los cursos por ciclo de los alumnos de una universidad.

1.1.1 Estructura propuesta:

La estructura propuesta, y cuyo uso será explicado en resultados posteriores se muestra a continuación:

```
Curso {
        String idCurso;
        String nombreCurso;
}

Usuario {
        String idEstudiante;
        String ciclo;
        String nombreEstudiante;
        List<Curso> cursos;
}
```

1.2 Resultado esperado 2: Modelo que permita recuperar el conocimiento previo del usuario a partir de una estructura en la que se encuentra codificada.

La recuperación del conocimiento a partir de la estructura en la que se encuentra codificada (base de datos relacional) se da a través de una única interacción cuando el usuario solicita explícitamente que se considere para la búsqueda la información respecto a los cursos que está llevando en el ciclo de consulta.

1.2.1 Funcionamiento

Para el diseño de este modelo se ha decidido que el hecho de considerar en el mecanismo de desambiguación al conocimiento del usuario sea opcional y decidido por el mismo usuario, ya que no necesariamente siempre éste querrá realizar búsquedas orientadas a la recuperación de información para sí mismo.

Este modelo está conformado por tres componentes.

En primer lugar, una capa de presentación que contiene la interfaz a través de la cual el usuario ingresará su consulta e indicará si desea incluir el conocimiento de sí mismo como criterio para la recuperación. En segundo lugar, una capa de negocio que

contiene la lógica completa para realizar la expansión. Dentro de esta lógica de expansión, el conocimiento del usuario cobrará relevancia al momento de realizar la desambiguación de palabras, ya que de darse el caso de que dos o más conceptos obtengan un mismo valor de similitud, el conocimiento del usuario podrá ser usado para orientar la solución a uno u otro resultado. A partir de este resultado, la lógica del cálculo de similitud se modifica ya que se agrega un flujo adicional, el cual consiste en verificar primero si el nodo analizado está contenido en el conocimiento del usuario, y de ser así el valor de similitud de dicho nodo se decrementará de tal forma que se acerque más al valor de similitud máximo.

Por otro lado, es en este mecanismo que cobra importancia la constante definida como CONSTANTE_REC mencionada en la sección 1.1.3 del capítulo 4, ya que de no existir esta holgura entre el valor de similitud de dos conceptos (holgura habilitada por dicha constante), podría darse el escenario de que al decrementar el valor de similitud de un concepto a causa del flujo adicional, se genere un nuevo empate de similitud con otro concepto, y este no es el objetivo de la estrategia de desambiguación por inclusión de conocimiento del usuario. Lo que busca la estrategia propuesta es marcar una diferencia entre dos conceptos que originalmente tenían un mismo valor de similitud, mas no generar una nueva ambigüedad.

Para realizar la verificación de si el concepto del nodo analizado está contenido en el conocimiento del usuario, se trabaja con los datos del usuario almacenados en un primer momento en la estructura planteada en el resultado anterior. Esta estructura fue inicialmente llenada de información a través de un componente Capa de Acceso a Datos que realizó la extracción de la información de la base de datos teniendo como único parámetro el código identificador del usuario.

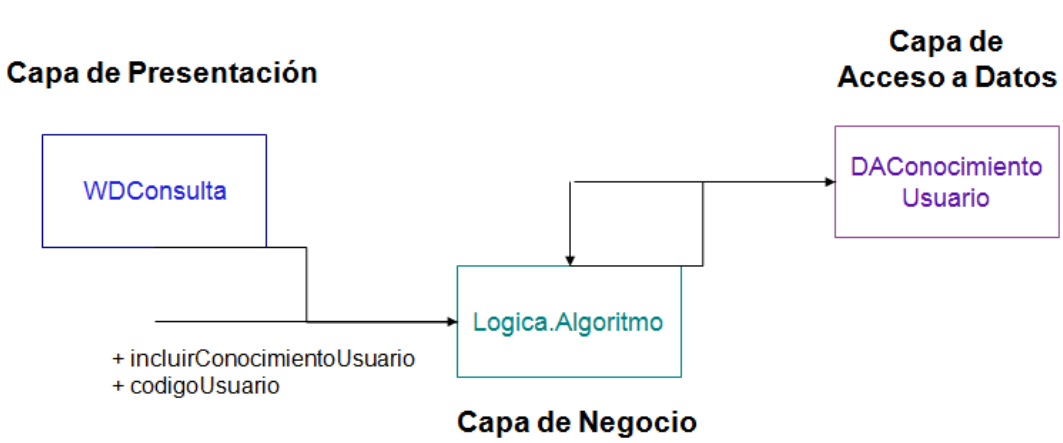


Imagen 11. Modelo de componentes para la recuperación del conocimiento del usuario.

1.2.2 Otras consideraciones

Para este proyecto, se ha optado por emplear el conocimiento del usuario para situaciones de desambiguación y no necesariamente incluir dichos términos directamente como términos para la expansión. Esto ya que la información obtenida como conocimiento del usuario, según la estructura para almacenar el conocimiento del usuario anteriormente expuesta, es más general. Adicionar dicho conocimiento como término de expansión en la totalidad de los casos podría generalizar la consulta y afectar el indicador de precisión.

1.2.3 Mecanismo de prueba

Para la verificación del mecanismo se realizó dos pruebas, la primera sin considerar el conocimiento del usuario:

Caso A

Prueba A1 – Sin considerar el conocimiento del usuario:

Input:

Incluir conocimiento del usuario activado: No	
Consulta inicial:	introduccion programacion
Consulta lematizada:	**introduccion programacion**

Output:

Palabra ambigua:	programacion
Palabra complementaria:	introduccion

***** RANKING *****

ID: ProgramacionC	**Lemma: programacion c**	**Valor: 16.0**
ID: ProgramacionOrientadaObjetos	**Lemma: Programacion orientar objeto**	**Valor: 16.0**
ID: LenguajesProgramacion	**Lemma: lenguaje Programacion 1**	**Valor: 16.0**
ID: FundamentosProgramacion	**Lemma: fundamento Programacion**	**Valor: 16.0**

Query expandido: programacion orientar objeto introduccion programacion fundamento programacion lenguaje programacion 1 programacion c

Prueba A2 - Considerando el conocimiento del usuario:

idCiclo	idCurso	Nombre	idEstudiante
2014-1	INF263	Algoritmia	20114316
2014-1	INF281	Lenguajes de Programacion 1	20114316

Imagen 12. Información de asignaturas cursadas por el usuario de prueba en el ciclo actual.

Input:

Incluir conocimiento del usuario activado: Sí	
Consulta inicial:	introduccion programacion
Consulta lematizada:	introduccion programacion

Output:

Palabra ambigua: programacion
Palabra complementaria: introduccion
Código: 20114316 Nombre: Carlos Arancivia

***** RANKING *****

ID: ProgramacionOrientadaObjetos Lemma: Programacion orientar objeto Valor: 14.0
ID: LenguajesProgramacion Lemma: lenguaje Programacion 1 Valor: 14.0
ID: ProgramacionC Lemma: programacion c Valor: 14.0
ID: FundamentosProgramacion Lemma: fundamento Programacion Valor: 16.0

Query expandido: programacion orientar objeto introduccion programacion lenguaje programacion 1 programacion c

Prueba A3 - Considerando el conocimiento del usuario:

Input:

Incluir conocimiento del usuario activado: Sí	
Consulta inicial:	programacion con objetos
Consulta lematizada:	programacion objeto

Output:

Palabra ambigua:	programacion	
Palabra complementaria:	objeto	
Código: 20114316	Nombre: Carlos Arancivia	
***** RANKING *****		
ID: ProgramacionOrientadaObjetos	**Lemma: Programacion orientar objeto**	**Valor: 0.0**
ID: ProgramacionC	**Lemma: programacion c**	**Valor: 14.0**
ID: LenguajesProgramacion	**Lemma: lenguaje Programacion 1**	**Valor: 14.0**
ID: FundamentosProgramacion	**Lemma: fundamento Programacion**	**Valor: 16.0**
Query expandido:	programacion orientar objeto objeto programacion	

En los resultados de la primera prueba se puede observar que todos los conceptos tienen un mismo nivel de similitud (máximo posible) ya que la palabra complementaria no fue encontrada en ningún nivel por lo que no contribuye a la desambiguación.

Sin embargo, para la segunda prueba se ha añadido y considerado los efectos de incluir el conocimiento del usuario en el proceso de expansión. Los resultados muestran que se ha logrado identificar una mejor opción sobre la otra.

Al analizar la ontología se puede observar que por transitividad “Programación Orientada a Objetos” es una unidad de aprendizaje perteneciente al curso de “Lenguajes de Programación 1”, y que la unidad de aprendizaje “Programación en C” pertenece al curso “Algoritmia”. Asimismo, en la imagen 12 se puede observar que el usuario de prueba está actualmente cursando “Algoritmia” y “Lenguajes de Programación 1” mas no “Fundamentos de Programación”, razón por la cual tiene sentido que las dos unidades de aprendizaje mencionadas y el mismo curso de Lenguajes de Programación hayan sido identificados como más apropiados entre las cuatro opciones presentes, mas no el nodo del curso Fundamentos de programación.

En el escenario de la prueba A2 no se cuenta con mayor información como para poder romper el empate entre los 3 términos identificados como mejores. De modificarse dicha consulta A2 reemplazando la palabra complementaria por otra más específica, se obtienen los resultados finales que se pueden observar en la prueba A3. Con esto se busca evidenciar que los mecanismos propuestos en el capítulo 4 y este se pueden complementar e interactúan apropiadamente para en conjunto generar mejores resultados.

1.3 Conclusiones

Este modelo se ha probado apropiado ya que cumple con el caso de prueba planteado con resultados lógicos según lo esperado.

CAPÍTULO 7

1 Objetivo específico 5: Diseñar un mecanismo que permita la recuperación por contenido semántico y no por contenido sintáctico.

Este objetivo busca exponer el último criterio para la obtención de términos de la consulta expandida, y con ello complementar el mecanismo anteriormente presentado que también estaba orientado a superar la diferencia entre cómo está expresado el conocimiento en el documento y cómo fue planteado en la consulta.

Adicional a ello, se presentará el mecanismo integrado de solución propuesto que permite la recuperación por contenido semántico considerando los beneficios individuales de cada componente expuesto en los objetivos anteriores.

1.1 Resultado esperado 1: Mecanismo de expansión empleando equivalencias de conceptos

Adicional a la expansión facilitada por el mecanismo manejador de ambigüedad, a continuación se plantea un nuevo mecanismo a considerar en el proceso de expansión.

1.1.1 Funcionamiento

Cada palabra de la consulta pre-procesada del usuario pasará por el mecanismo de detección de equivalencias del componente EquivalenceHandler.

Dicho mecanismo de detección de equivalencias consiste en que inicialmente se recuperará en una lista el conjunto de nodos que no contengan en dicho nodo raíz al concepto analizado. A continuación, se buscará en cada uno de los sinónimos de dicho nodo la presencia del concepto analizado y si efectivamente ese concepto es encontrado en uno de los sinónimos automáticamente todas las palabras relacionadas a ese nodo incluyendo los sinónimos serán añadidos como términos para la expansión.

1.1.2 Mecanismo de prueba

Para la verificación del mecanismo se realizó dos pruebas, la primera sin considerar este nuevo mecanismo:

Caso A

Prueba A1 – Sin considerar el mecanismo de expansión por equivalencias:

Input:

Consulta inicial:	programacion poo
Consulta lematizada:	**programacion poo**

Output:

Incluir conocimiento del usuario activado: No

Palabra ambigua: programacion

Palabra complementaria: poo

***** RANKING *****

ID: ProgramacionC	**Lemma: programacion c**	**Valor: 16.0**
ID: ProgramacionOrientadaObjetos	**Lemma: Programacion orientar objeto**	**Valor: 16.0**
ID: LenguajesProgramacion	**Lemma: lenguaje Programacion 1**	**Valor: 16.0**
ID: FundamentosProgramacion	**Lemma: fundamento Programacion**	**Valor: 16.0**

Prueba A2 - Segunda prueba considerando el mecanismo de expansión por equivalencias:

Input:

Consulta inicial:	programacion poo
Consulta lematizada:	**programacion poo**

Output:

Incluir conocimiento del usuario activado: No

Palabra ambigua: programacion

Palabra complementaria: poo

***** RANKING *****

ID: ProgramacionOrientadaObjetos	**Lemma: Programacion orientar objeto**	**Valor: 47.0**
ID: ProgramacionC	**Lemma: programacion c**	**Valor: 61.0**
ID: LenguajesProgramacion	**Lemma: lenguaje Programacion 1**	**Valor: 61.0**
ID: FundamentosProgramacion	**Lemma: fundamento Programacion**	**Valor: 61.0**

Query expandido: programacion poo programacion orientar objeto oop

A partir del resultado de la primera prueba se puede observar que todos los conceptos obtenidos han tenido un mismo nivel de similitud el cual a su vez es el valor más alejado a lo que podría considerarse equivalente o similar ya que la palabra complementaria no ha servido para la identificación del concepto apropiado.

Sin embargo, en la segunda prueba para la cual sí se empleó el mecanismo que considera la existencia de las equivalencias o palabras sinónimas de los conceptos, ya se logró una correcta identificación de los mejores conceptos a añadir como nuevos términos en la consulta expandida. En esta prueba en particular se puede apreciar ya que la ontología tiene mapeada la relación de equivalencia entre los conceptos de "POO" y "Programación Orientada a Objetos".

1.2 Resultado esperado 2: Modelo de interpretación y conversión de la consulta del usuario usando ontologías para la interpretación y expansión de consultas para la conversión.

A continuación se presenta el modelo integrado propuesto y compuesto por los distintos subcomponentes creados a lo largo de los resultados de cada objetivo.

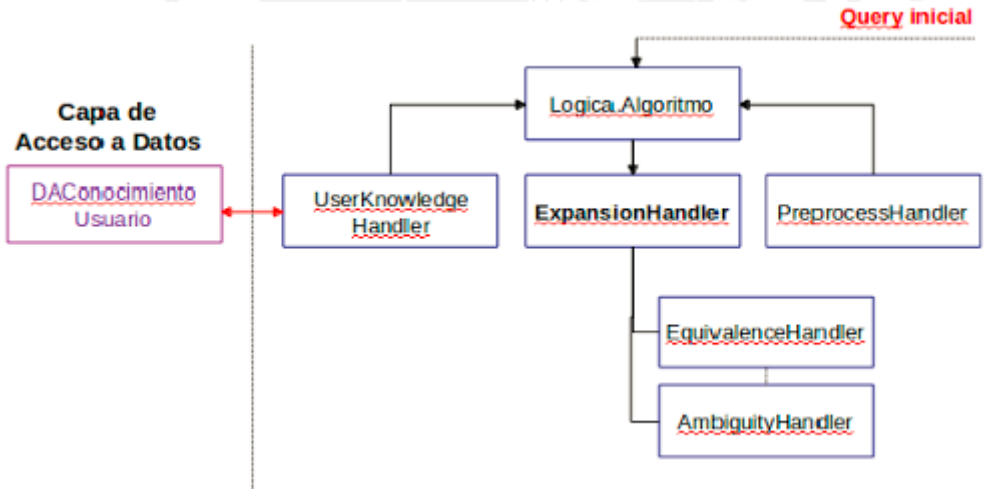


Imagen 13. Modelo integrado de componentes

1.2.1 Funcionamiento

En primer lugar, este mecanismo emplea la consulta inicial provista por el usuario. A continuación, esta consulta pasa a un componente lógico orquestador denominado Algoritmo. Este orquestador invoca al componente *PreprocessHandler* que se encarga de obtener una consulta pre-procesada, sin *stopwords* y con términos en su forma base.

En seguida se evalúa si el usuario solicitó incluir su información como factor en el proceso de búsqueda. De ser así el orquestador invoca al componente *UserKnowledgeHandler* quien solicita al componente de acceso a datos del usuario recuperar la información de dicho usuario para su posterior uso cuando sea requerido por el *AmbiguityHandler*.

Continuando con el flujo, la consulta pre-procesada devuelta por el *PreprocessHandler* es enviada al componente *ExpansionHandler*, el cual realiza en un primer momento la expansión por equivalencias a través del componente *EquivalenceHandler*. Los nuevos términos obtenidos en base a la expansión por equivalencia son grabados en una consulta denominada Query Expandido.

A continuación, cada uno de los términos de dicho Query Expandido (incluyendo los obtenidos por equivalencia) pasa por un proceso de desambiguación contenido en el componente *AmbiguityHandler*, tras lo cual se obtienen los términos finales para la consulta expandida.

1.2.2 Mecanismo de prueba

Como mecanismo de prueba de la solución integrada propuesta se empleará el modelo usado en el siguiente resultado.

1.3 Resultado esperado 3: Modelo que permita la recuperación de información relevante dada una consulta.

A continuación se presenta el modelo añadiendo el módulo que realiza el proceso de recuperación de información en base a la consulta expandida.

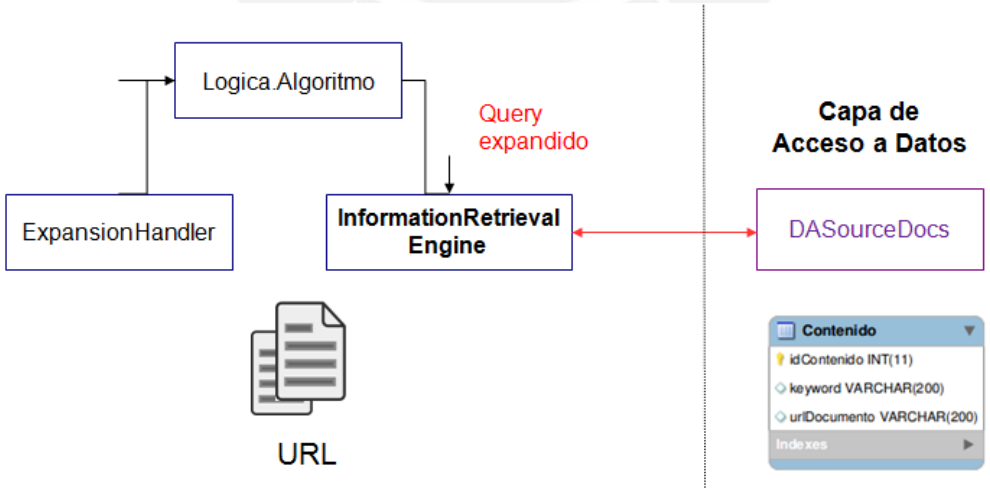


Imagen 14. Modelo con integración del componente del motor de recuperación.

1.3.1 Funcionamiento

Para efectos de lograr la recuperación de información se ha añadido un componente denominado *InformationRetrievalEngine* el cual recibe como input la consulta expandida, y en base a esta consulta y a los documentos etiquetados que se encuentran en una base de datos realiza la recuperación.

Para este resultado se ha configurado un motor de recuperación basado en Lucene sin ahondar en el diseño de uno propio.

1.3.2 Mecanismo de prueba

Para la verificación del mecanismo se realizó dos pruebas sobre una misma consulta.

Caso A

Prueba A1 – Considerando el mecanismo de desambiguación sin apoyo de palabras complementarias:

Input:

Consulta inicial:	pilas
Consulta lematizada:	**pila**

Output:

Incluir conocimiento del usuario activado: No
Palabra ambigua: pila
Palabra complementaria: -
***** RANKING *****

ID: AplicacionesPilas	Lemma: aplicación pila	Valor: 0.0	Nivel: 1
ID: TADPilas	Lemma: tipo abstracto pila	Valor: 0.0	Nivel: 1

Query expandido: tipo abstracto pila pila aplicación pila

Prueba A2 – Considerando el mecanismo de uso del conocimiento del usuario para la desambiguación:

Input:

Consulta inicial:	pilas

Consulta lematizada:	**pila**

Output:

Incluir conocimiento del usuario activado: Sí
Palabra ambigua: pila
Palabra complementaria: -
Código: 20114316 Nombre: Carlos Arancivia

***** RANKING *****

ID: AplicacionesPilas	Lemma: aplicación pila	Valor: -1.0	Nivel: 1
ID: TADPilas	Lemma: tipo abstracto pila	Valor: 0.0	Nivel: 1

Query expandido: pila aplicación pila

Debido a que el mecanismo de desambiguación empleando el conocimiento del usuario introduzco un flujo adicional a través del que es posible que se decremente el valor de similitud, el valor de similitud máximo puede llegar a ser -1 respecto al máximo definido por el flujo regular (0). Es decir, siendo 0 el valor del máximo definido para el flujo regular, el valor máximo real puede llegar a ser hasta -1.

Prueba A3 – Considerando el mecanismo de expansión por equivalencia:

Input:

Consulta inicial:	pilas y quicksort
Consulta lematizada:	**pila quicksort**

Output:

Incluir conocimiento del usuario activado: Sí
Palabra ambigua: programacion
Palabra complementaria: poo
Código: 20114316 Nombre: Carlos Arancivia

***** RANKING *****

ID: AplicacionesPilas	Lemma: aplicación pila	Valor: 42.0	Nivel: 1
ID: TADPilas	Lemma: tipo abstracto pila	Valor: 46.0	Nivel: 1

Query expandido: pila ordenamiento rápido aplicación pila quicksort

En primer lugar, el mecanismo de pre-procesamiento recibió la consulta inicial y procedió con la remoción del *stopword* "y" y la lematización de los *tokens* ("pilas" por "pila"). A continuación, la consulta lematizada ingresó al componente Algoritmo, el cual es el componente principal (orquestador). Lo primero que realizó dicho componente es verificar si el usuario solicitó emplear su información en la búsqueda, y como para este caso de prueba efectivamente se solicitó, transfirió al componente de acceso a datos del usuario la información del código del usuario para recibir en respuesta la información completa de cursos del usuario en el ciclo actual a través de una consulta a la base de datos. Luego, el orquestador envió la consulta al componente ExpansionHandler, el cual a su vez derivó la misma al componente EquivalenceHandler para verificar si alguno de los términos de la consulta pre-procesada está en algún elemento sinónimo (pero no en los elementos de propiedad *lemma*) de la ontología, para así evitar que queden excluidos de ser revisados en el siguiente paso. En el caso de este ejemplo, la palabra *Quicksort* se encontró como sinónimo (término equivalente) del nodo "Ordenamiento rápido", por lo que dicho nodo se agregó como término de expansión. La relevancia de este término para efectos de la expansión finaliza aquí ya que *quicksort* no es ambiguo.

Por otro lado, el *token* "pila" también sigue los mecanismos anteriores sin mayor relevancia respecto al EquivalenceHandler, ya que "pila" no está mapeado como equivalencia de algún término. Continuando con el flujo, el componente AmbiguityHandler se encarga de recuperar todas las coincidencias del *token* analizado en la propiedad *lemma,* y calcula los valores de similitud entre el *token* ambiguo y el complementario. Para el caso de ejemplo, a raíz de la ambigüedad del término "pila" se debería obtener tanto el nodo TADPilas y AplicacionesPilas tienen un mismo nivel de similitud óptima. En este punto cobra relevancia la información del usuario, ya que como se ve en la descripción de la ontología en la sección 1.1.1 del capítulo 5, el nodo AplicacionesPilas pertenece, por transitividad, al curso Algoritmia que el estudiante sí está llevando en el ciclo actual, sin embargo, el nodo TADPilas pertenece a Fundamentos de Programación, curso que el estudiante no está llevando actualmente. Es por eso que se obtuvo como término más apropiado "Aplicación pilas" y no "tipo abstracto pila" a pesar de que ambos contienen el término ambiguo "pila".

Para efectos de las pruebas de recuperación de información según la consulta expandida se ha etiquetado un total de 30 evaluaciones y almacenado en la base de datos que presenta la siguiente estructura:

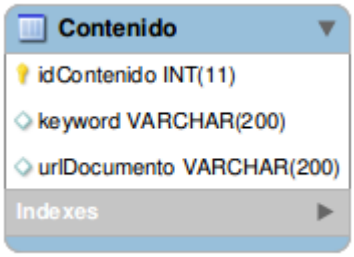


Imagen 15. Estructura de la tabla de la base de datos que almacena los datos de los documentos que conforman las fuentes de información

Para proceder con la evaluación de los resultados, en las tablas a continuación se ha colocado los resultados de la búsqueda del usuario ordenados por relevancia sin emplear el mecanismo de expansión y empleando el mecanismo de expansión:

Query preprocesado: pila quicksort

Score	URI	Etiquetas
0.224284	http://tesis.bcc.com/2013-2/AL/L1	/ aplicación estructura dato / aplicación cola / aplicación **pila** /
0.149522	http://tesis.bcc.com/2013-2/FP/PC2	/ derivacion programa / lenguaje ordenar guarda / Especificacion algebraico tipo dato abstracto / tipo abstracto **pila** /
0.149522	http://tesis.bcc.com/2013-2/FP/PC3	/ Especificacion algebraico tipo dato abstracto / tipo abstracto conjunto / tipo abstracto **pila** /
0.149522	http://tesis.bcc.com/2013-1/FP/PC3	/ derivacion programa / calcular predicado / Especificacion algebraico tipo dato abstracto / tipo abstracto **pila** / tipo abstracto lista /
0.149522	http://tesis.bcc.com/2013-2/AL/L2	/ aplicación lista / aplicación **pila** / aplicación estructura dato / aplicación arbolar / Ordenacion / Ordenamiento rapido /
0.130832	http://tesis.bcc.com/2013-1/FP/PC4	/ Especificacion algebraico tipo dato abstracto / tipo abstracto conjunto / tipo abstracto **pila** / tipo abstracto cola / tipo

		abstracto vector / tipo abstracto grafo /

Tabla 13. Documentos recuperados empleando solo el *query* pre-procesado (sin expansión).

Query expandido: pila ordenamiento rápido aplicación pila quicksort

Score	URI	Etiquetas
1.27672	http://tesis.bcc.com/2013-2/AL/L2	/ aplicación lista / **aplicación pila** / **aplicación estructura dato** / aplicación arbolar / Ordenacion / **Ordenamiento rapido** /
0.57709	http://tesis.bcc.com/2013-2/AL/L1	/ **aplicación estructura dato** / aplicación cola / **aplicación pila** /
0.29323	http://tesis.bcc.com/2013-1/AL/L4	/ Busqueda / tratamiento colisionar / Ordenacion / **ordenamiento rapido** / Ordenacion selección /
0.129657	http://tesis.bcc.com/2013-2/FP/PC2	/ derivacion programa / lenguaje ordenar guarda / Especificacion algebraico tipo dato abstracto / tipo abstracto **pila** /
0.129657	http://tesis.bcc.com/2013-2/FP/PC3	/ Especificacion algebraico tipo dato abstracto / tipo abstracto conjunto / tipo abstracto **pila** /
0.1296565	http://tesis.bcc.com/2013-1/FP/PC3	/ derivacion programa / calcular predicado / Especificacion algebraico tipo dato abstracto / tipo abstracto **pila** / tipo abstracto lista
0.113449	http://tesis.bcc.com/2013-1/FP/PC4	/ Especificacion algebraico tipo dato abstracto / tipo abstracto conjunto / tipo abstracto **pila** / tipo abstracto cola / tipo abstracto vector / tipo abstracto grafo /
0.09061	http://tesis.bcc.com/2013-1/AL/L1	/ **aplicación estructura dato** / **aplicación** lista /
0.064722	http://tesis.bcc.com/2013-1/AL/L2	/ **aplicación estructura dato** / **aplicación** arbolar / Busqueda / Busqueda secuencial /
0.051777	http://tesis.bcc.com/2013-2/AL/L5	/ Busqueda / tratamiento colisionar / Ordenacion / **Ordenacion intercambiar** /

		Ordenacion monticulos / **aplicación estructura dato** / aplicación cola /

Tabla 14. Documentos recuperados empleando el *query* expandido según el modelo propuesto.

Para el caso de ejemplo, si bien el usuario no detalló su consulta, se dedujo que el usuario al solicitar que se incluyera su información en la búsqueda quiso orientar las respuestas al tema de aplicaciones de estructuras de datos, y en particular a las aplicaciones del tipo de estructura pilas, y adicionalmente, en lo posible que también se incluyeran ejercicios relacionados al método *quicksort*. En la primera prueba lo recuperado fue la totalidad de coincidencias del término pila en los documentos etiquetados, y los de más altos puntajes (los que deberían ser los más relevantes para el usuario) son en su mayoría los relacionados a la teoría de tipos abstractos de datos del curso fundamentos de programación. En los últimos lugares está el documento que realmente le era relevante (que incluye ambos temas en una misma evaluación). De las 6 respuestas solamente dos de ellas son relevantes, lo cual da una precisión de 33%.

Por otro lado, en los resultados empleando expansión de consultas, el primer documento recuperado corresponde al más relevante (que incluye ambos temas en una misma evaluación), y paulatinamente se van brindando otros documentos que son relevantes en menor grado (ya que incluyen ejercicios de aplicaciones de la estructura pilas, ejercicios de aplicaciones de estructuras análogas o ejercicios de *Quicksort*). Este análisis resulta en que de los 10 documentos recuperados 7 tienen un grado de relevancia para el usuario, lo cual da una precisión del 70%, mayor que sin el uso de la técnica de expansión.

A continuación se muestra la tabla con el resumen de los resultados de los casos expuestos a lo largo del presente trabajo:

Consulta inicial	Sin expandir	Con expansión	Mecanismo relevante	Indicador de precisión
Pilas	pila			2/6 = 0.33
		pila aplicación pila	Conocimiento del usuario	5/9 = 0.56
Archivos algoritmia	archivo algoritmia			0/5 = 0
		algoritmia archivo ordenacion archivo	Desambiguación por palabra complementaria	5/10 = 0.5
poo	poo			0/4 = 0
		poo programacion orientar objeto oop	Equivalencias	4/4 = 1
Pilas y quicksort	Pila quicksort			2/6 = 0.33
		pila ordenamiento rápido aplicación pila quicksort	Conocimiento del usuario, equivalencias	6/10 = 0.70

Tabla 15. Resumen de resultados con medición de precisión.

1.4 Conclusiones

Para este objetivo, se ha observado que se ha logrado integrar adecuadamente cada uno de los componentes definidos en los objetivos anteriores a una única propuesta de solución final, que partiendo de una consulta genérica y posiblemente ambigua, logra recuperar información relevante para el usuario y superando las diversas causas identificadas como raíz de la problemática. Ello también fue verificado con la medición y comparativa de los resultados según el indicador de precisión. Como se vio, la solución propuesta mejora los resultados en un 37%.

Como conclusión final se pudo observar que el conjunto de mecanismos propuestos de forma independiente funcionan adecuadamente para generar términos de expansión apropiados, y de forma integrada proveen una mejora significativa ante los escenarios que se derivan de la problemática inicial del presente proyecto de fin de carrera.

CAPÍTULO 8

1 Conclusiones

A lo largo del trabajo expuesto se observa que se ha logrado una correcta integración de cada uno de los componentes a una única propuesta de solución final, ya que partiendo de una consulta genérica y posiblemente ambigua, se logra recuperar información relevante para el usuario, superando a cierto grado las dificultades identificadas como raíz de la problemática. Para ello, se ha realizado pruebas de cada mecanismo de forma individual y a través de una evaluación cualitativa, identificado que efectivamente se obtienen mejoras en los resultados (sin expansión – con expansión). Las consultas planteadas para las pruebas fueron propuestas considerando aquellas que fueran mejores exponentes de las características de ambigüedad y diversidad de representaciones de un mismo concepto del lenguaje natural.

Bajo estas condiciones, los resultados fueron medidos con un indicador de “Precisión”. Las pruebas sin expansión dieron un overall de precisión de 16.5% mientras que las pruebas con expansión dieron un overall de precisión de 69%, con lo que se reafirmó que se obtuvo mejoras en los resultados.

2 Recomendaciones y trabajos futuros

Como trabajo futuro se propone el escalamiento del proyecto hacia un modelo que permita la recuperación de información de otros dominios de conocimiento diferentes al empleado en este proyecto a través del reemplazo de la ontología, de tal forma que se pueda ampliar los beneficiarios de la propuesta.

Asimismo, se propone la integración de la herramienta con mecanismos de extracción automática de información de fuentes online, de tal forma que las fuentes de información sobre las cuales realizar la búsqueda sea más amplio y dinámico.

También se propone la inclusión de un mecanismo automático de captura de información del usuario, para propiciar la obtención de correctos resultados que vayan acorde a la realidad dinámica del usuario.

Finalmente, se propone la inclusión de módulos de análisis y pre-procesamiento de la consulta inicial del usuario del usuario por lógica proposicional.

REFERENCIAS BIBLIOGRÁFICAS

[ALI, KHAN 2008] ALI, Waris; KHAN, Sharifullah.
2008 Ontology driven query expansion in data integration. Semantics, Knowledge and Grid, 2008. SKG'08. Fourth International Conference on. Beijing, 2008. pp. 57-63.

[APACHE, 2011] APACHE SOFTWARE FOUNDATION
2011 “Lucene Features”. Apache Lucene Core. 2011. Consulta: 5 de noviembre de 2013.
<http://lucene.apache.org/core/index.html>

[APACHE, 2011] APACHE SOFTWARE FOUNDATION
2011 “Getting started with Apache Jena”. Apache Jena. 2011. Consulta: 15 de noviembre de 2013.
<http://jena.apache.org/getting_started/index.html>

[BAEZA-YATES, RIBERIRO-NETO 1999] BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier.
1999 Modern information retrieval. 1st Edition. Boston: Addison-Wesley Longman Publishing Co., Inc.

[BAI, NIE 2007] BAI, Jing; NIE, Jian-Yun; CAO, Guihong; BOUCHARD, Huques
2007 "Using query contexts in information retrieval." Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. New York, 2007, pp. 15-22.

[BUCKLAND apud BHOGAL, 2007; NAVIGLI, BELARDI apud BHOGAL, 2007; BHOGAL, 2007; BHOGAL, MACFARLANE et al 2007] BHOGAL, Jagdev; MACFARLANE, Andy; SMITH, Peter.
2007 “A review of ontology based query expansion”. Information processing and management. New York, 2007, vol. 43, no 4, p. 866-886.

[CALEGARI, SANCHEZ 2008] CALEGARI, Silvia; SANCHEZ, Elie.
2008 “Object-fuzzy concept network: An enrichment of ontologies in semantic information retrieval.” Journal of the American Society for Information Science and Technology, 2008, vol. 59, no 13, pp. 2171-2185

[CALEGARI, PASI 2008] CALEGARI, Silvia; PASI, Gabriella.
2008 “Personalized ontology-based query expansion.” Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on. Milan, 2008, vol. 13, p. 256-259.

[CERULO, CANFORA 2004] CERULO, Luigi; CANFORA, Gerardo.
2004 “A taxonomy of information retrieval models and tools”. Journal of Computing and Information Technology. 2004, vol. 12, no 3, pp. 175-194.

[CHEN, et al 2012] CHEN, Huacheng; DU, Xuehui; CHEN, Xingyuan; XIA, Chuntao.
2012 “Query expansion model based on interest ontology.” Information Management, Innovation Management and Industrial Engineering (ICIII), 2012 International Conference on. 2012, vol. 3, pp. 474-478.

[DEY, et al 2005] DEY, Lipika; SINGH, Shailendra; RAI, Romi; GUPTA, Saurabh.
2005 “Ontology aided query expansion for retrieving relevant texts.” Advances in Web Intelligence. Berlin, 2005, pp. 126-132.

[FU, et al 2005] FU, Gaihua; JONES, Christopher; ABDELMOTY, Alia.
2005 “Ontology-based spatial query expansion in information retrieval.” On the move to meaningful Internet systems 2005: CoopIS, DOA, and ODBASE. Berlin, 2005. pp. 1466-1482.

[GENESERETH & NILSSON apud GRUBBER, 1995] GRUBER, Thomas.
1995 “Toward principles for the design of ontologies used for knowledge sharing?” International journal of human-computer studies, 1995, vol. 43, no 5, pp. 907-928.

[GREENGRASS, 2009] GREENGRASS, Ed.
2000 Information retrieval: A survey. University of Maryland Publishing.

[INGWERSEN, 1992] INGWERSEN, Peter.
1992 Information Retrieval Interaction. Taylor Graham Publishing.

[JALALI, BORUJERDI 2008] JALALI, Vahid; BORUJERDI, Mohammad.

2008 “The effect of using domain specific ontologies in query expansion in medical field.” Innovations in Information Technology, 2008. IIT 2008. International Conference on. Al Ain, 2008, pp. 277-281.

[KEVIN, 2010] KEVIN, Kelly
2010 What Technology Wants. New York: Viking Press.

[MAMBO, 2004] MAMBO FOUNDATION
2004 “Features”. Freeling Features. 2004. Consulta: 5 de noviembre de 2013. <http://nlp.lsi.upc.edu/freeling/index.php?option=com_content&task=view&id=12&Itemid=41>

[MITRA, CHAUDHURI 2000] MITRA, Mandar; CHAUDHURI, B. B.
2000 “Information retrieval from documents: A survey”. Information Retrieval. 2000, vol. 2, no 2-3, pp. 141-163.

[MITRA, SINGHAL et al 2000] MITRA, Mandar; SINGHAL, Amit; BUCKLEY, Chris.
1998 “Improving automatic query expansion”. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. Melbourne, 1998, pp. 206-214.

[PAHA, GULATI et al 2009] PAHA, Nisha; GULATI, Payal; GUPTA, Parul.
2009 Ontology driven conjunctive query expansion based on mining user logs. Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on. Delhi, 2009. pp. 1-4.

[PARALIC, KOSTIAL 2003] PARALIC, Jan; KOSTIAL, Ivan
2003 "Ontology-based information retrieval". Information and Intelligent Systems. Croatia, pp. 23-28.

[RIJSBERGEN, 1979] RIJSBERGEN, Joost.
1979 Information Retrieval. 2nd Edition. London: Butterworths.

[SALTON, 1981] SALTON, Gerard.
1981 "A blueprint for automatic indexing". SIGIR Forum. 1981, vol. 16, no 2, pp. 22-38.

[SENDHILKUMAR, GEETHA 2008] SENDHILKUMAR, S.; GEETHA, T. V.

2008 “Personalized ontology for web search personalization.” Proceedings of the 1st Bangalore annual Compute conference. New York, 2008, no 18.

[SINGHAL, 2001] SINGHAL, Amit.
2001 "Modern information retrieval: A brief overview". IEEE Data Eng. Bull. 2001, vol. 24, no 4, pp. 35-43.

[SONG, et al 2005] SONG, Min; SONG, Il-Yeol; HU, Xiaohua; ALLEN, Robert
2005 "Semantic query expansion combining association rules with ontologies and information retrieval techniques." Data Warehousing and Knowledge Discovery. Berlin, 2005, pp. 326-335.

[STANFORD, 2013] STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH
2014 “What’s Protégé?”. California, 2013. Consulta: 5 de noviembre de 2013.
<http://protege.stanford.edu/overview/>

[W3C, 2009] WORLD WIDE WEB CONSORTIUM
2009 “OWL Overview”. Consulta: 14 de noviembre de 2013.
<http://www.w3.org/TR/owl-features/>

[W3C, 2014] WORLD WIDE WEB CONSORTIUM
2014 “SPARQL Query Language for RDF”. Consulta: 20 de marzo de 2014.
< http://www.w3.org/TR/rdf-sparql-query/ >

[WANG, et al 2009] WANG, Hongsheng; QIN, Jiuying; SHAO, Hong
2009 "Expansion model of semantic query based on ontology." Web Mining and Web-based Application 2009. WMWA'09. Second Pacific-Asia Conference on. Wuhan, 2009, pp. 86-90

[WANG, et al 2012] WANG, Haoming; GUO, Ye; SHI, Xibing; YANG, Fan
2012 "Conceptual representing of documents and query expansion based on ontology." Web Information Systems and Mining. Chengdu, 2012, vol. 7259, pp. 489-496.

[KITCHENHAM, 2004] KITCHENHAM, Barbara
2004 "Procedures for performing systematic reviews" Keele, 2004, vol. 33, pp. 2004.

[LU, et al 2013] LU, Zhao; YAN, Zhixian; HE, Liang
2013 "OnPerDis: Ontology-Based Personal Name Disambiguation on the Web." Web Intelligence (WI) and Intelligent Agent Technologies (IAT). IEEE/WIC/ACM International Joint Conferences. 2013, vol. 1, pp. 185-192.