PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DESARROLLO DE UNA HERRAMIENTA QUE PERMITA LA EXTRACCIÓN DE UNA TAXONOMÍA DE UN CONJUNTO DE DOCUMENTOS DE UN DOMINIO ESPECÍFICO USANDO CFINDER PARA LA EXTRACCIÓN DE CONCEPTOS CLAVE

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

Alfredo Adrián Vargas Rosales

ASESOR: Héctor Andrés Melgar Sasieta

Lima, febrero de 2015

Resumen

Gracias a la *World Wide Web* la idea de información compartida alrededor del mundo es común para todos, la información es ingresada desde diferentes fuentes para que todos puedan verla y usarla. Una *Smart Web* o *Semantic Web* tiene como objetivo estructurar los contenidos de forma tal que todo esté relacionado y por lo tanto, presente información consistente. Para ello, se requieren de estructuras que puedan ser accesadas por computadoras y contengan reglas de inferencia para un razonamiento automático. Una de estas estructuras es la ontología. Una ontología busca conceptualizar el conocimiento de un dominio específico valiéndose de representaciones. Como primer paso para construir una ontología, se debe obtener una taxonomía.

Una taxonomía es una clasificación de entidades de información a manera de jerarquías. Las taxonomías ofrecen diversas ventajas como clasificar de la información, realizar búsquedas de manera más eficaz y navegar entre muchos conceptos, sin embargo, requieren mucho esfuerzo para ser construidas a mano. Para poder construir una taxonomía en base a un grupo de documentos, primero se debe extraer los conceptos más relevantes presentes en dichos textos. Luego, se debe deducir la jerarquía se convertirá en la taxonomía.

Para extraer los conceptos más relevantes de un grupo de documentos, el método *CFinder* ha probado ser muy útil y dar buenos resultados. El objetivo del *CFinder* es que sea usado para la construcción de ontologías u otro tipo de estructura que requiera una fase de extracción de conceptos clave. No obstante, no se ha integrado con un método que permita estructurar la jerarquía entre los conceptos extraídos.

En este proyecto se busca complementar el método *CFinder* con una fase para la deducción de jerarquías entre los conceptos extraídos y la construcción de una taxonomía, de esta manera, se está brindando una nueva opción para la construcción automática de taxonomías. Para ello, se realiza la implementación de una herramienta para la construcción automática de una taxonomía de un dominio que haga uso del método *CFinder*.

El proyecto inicia con la implementación de un módulo que permite la extracción de conceptos clave de un conjunto de documentos usando el método *CFinder*. Luego, se procede a implementar un módulo que permita extraer una taxonomía usando los conceptos clave extraídos. Finalmente, se realizan las pruebas necesarias para medir la eficacia del método implementado y, con los resultados obtenidos, se concluye que se alcanzó el objetivo principal del proyecto.

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: **Desarrollo de una herramienta que permita la extracción de una taxonomía de un conjunto de documentos de un dominio específico usando CFinder para la extracción de conceptos clave**

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Facultad de Ciencias e Ingeniería
09 ABR. 2015
MIGUEL MEJIA PUENTE
DECANO

ÁREA: Ciencias de la computación
PROPONENTE: Dr. Héctor Andrés Melgar Sasieta
ASESOR: Dr. Héctor Andrés Melgar Sasieta
ALUMNO: Alfredo Adrián Vargas Rosales
CÓDIGO: 20095526
TEMA N°: 579
FECHA: San Miguel, 14 de diciembre del 2014

DESCRIPCIÓN

Las taxonomías ofrecen diversas ventajas como clasificar información, realizar búsquedas de manera más eficiente y navegar entre muchos conceptos, sin embargo, requieren mucho esfuerzo para ser construidas manualmente. Para poder construir una taxonomía en base a un grupo de documentos, primero se debe extraer los conceptos más relevantes presentes en dichos textos. Luego, se debe deducir la jerarquía que, finalmente se convertirá, en la taxonomía.

Para extraer los conceptos más relevantes, el método CFinder ha probado ser muy útil y superior a otros métodos, no obstante, no se ha integrado con un método que permita estructurar la jerarquía entre los conceptos extraídos. Por este motivo, se propone la implementación de una herramienta para la construcción automática de una taxonomía de un dominio que haga uso del método CFinder. Con esta herramienta, se aportará un nuevo método de extracción de una taxonomía en base a un conjunto de documentos.

OBJETIVO GENERAL

Implementar una herramienta que soporte la construcción de manera automática de una taxonomía en un dominio a partir de un conjunto de textos.

OBJETIVOS ESPECÍFICOS

- Desarrollar un subsistema que permita la extracción automática de conceptos claves.
- Desarrollar un subsistema que permita la identificación de jerarquías y construcción de una taxonomía de forma automática.
- Comprobar el correcto funcionamiento de la herramienta realizando las pruebas necesarias a las taxonomías generadas.

Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono:
(511) 626 2000 Anexo 4801

ALCANCE

El proyecto se encuentra dentro del área de las ciencias de la computación, específicamente en la ingeniería del conocimiento. El proyecto busca la implementación de una herramienta para la construcción automática de una taxonomía en base a un corpus de documentos en el lenguaje inglés dentro de un dominio específico del documento. Además, se usará el método CFinder para la extracción de los conceptos clave y se busca añadir a este método la fase de extracción de jerarquías y construcción de una taxonomía. Al momento de realizar la extracción de palabras clave, se requiere un glosario de términos específicos del dominio el cual será complicado encontrar para todos los dominios. El diseño de este glosario de términos no es parte de la solución. Los documentos usados como fuentes para la construcción de la taxonomía deberán ser de un mismo dominio, ya que la herramienta buscará armar solo una única taxonomía de todos los documentos.

Máximo: 100 páginas

Av. Universitaria 1801
San Miguel, Lima – Perú

Apartado Postal 1761
Lima 100 – Perú

Teléfono:
(511) 626 2000 Anexo 4801

A mis padres Alicia y Alfredo quienes siempre me han dado todo, nunca en mi vida me faltó nada gracias a su arduo trabajo y esfuerzo. A mi mami María y mi tati Eloy quienes en mis veintidós años de vida siempre se han preocupado y cuidado de mí.

AGRADECIMIENTOS

A mi asesor Andrés Melgar por su constante apoyo, sus consejos y guía durante el desarrollo del proyecto. A toda mi familia quienes siempre han estado presente para darme las fuerzas de seguir adelante. A mi centro laboral, Dirinfo, por concederme siempre el tiempo que necesitaba para trabajar en mi proyecto de tesis y a los miembros de mi equipo por todo el apoyo brindado.

Contenido

Índice de tablas

Índice de imágenes

1 Capítulo 1

Las taxonomías nos ofrecen distintas ventajas, pero se requiere gran esfuerzo y recursos para construirlas. Primero, se requiere extraer las palabras más importantes y posteriormente, encontrar las jerarquías entre ellas. Para realizar la primera tarea existe un método conocido como CFinder, al cual, no se le ha implementado una fase de extracción de jerarquías. Es por ello que en este proyecto se propone desarrollar una herramienta para la construcción de manera automática de la taxonomía de un dominio a partir de textos el cual usará CFinder para extraer los conceptos relevantes. De esta manera, se obtendrá una nueva herramienta y un nuevo método para la extracción de taxonomías a partir de textos.

1.1 Problemática

Gracias a la *World Wide Web* la idea de información compartida alrededor del mundo es común para todos, la información es ingresada desde diferentes fuentes para que todos puedan verla y usarla [19]. Por este motivo, cuando un usuario busca información, espera que esta sea consistente, es decir, que la información presentada tenga relación entre sí. Sin embargo, esto no siempre ocurre debido a que mucha de la *data* que se presenta en la web no está relacionada entre sí, ni siquiera dentro de un mismo sitio web; a este hecho se le conoce como *Dumb Web* [19]. Por el contrario, una *Smart Web* o *Semantic Web* tiene como objetivo estructurar los contenidos de forma tal que todo esté relacionado. Esto permite que la información se mantenga actualizada, pues para modificar la *data* en todos los lugares donde esté relacionada, solo es necesaria cambiarla una sola vez. Además facilita la búsqueda de información, pues los datos se encuentran interrelacionados [19].

Para almacenar y relacionar los datos, se requieren de estructuras semánticas que puedan ser accedidas por computadoras y contengan reglas de inferencia para un razonamiento automático, la idea principal no es solo enlazar documentos entre sí, sino reconocer el significado de la información que almacena [22]. Una de estas estructuras es la ontología. Una ontología busca conceptualizar el conocimiento de un dominio específico valiéndose de representaciones como los objetos, clases, relaciones [2] [3]. También puede definirse una ontología como todo documento o archivo que define formalmente las relaciones entre términos [22]. Aunque tiene muchos significados, el conocimiento puede ser definido como un estado de la mente, un objeto, un proceso, una capacidad de acceder a la información [24]. La adquisición del conocimiento requerido para la construcción de una ontología normalmente toma mucho tiempo y recursos [1], por este motivo, se han buscado diferentes formas de automatizar su construcción. A los métodos usados para disminuir los recursos requeridos para la construcción de una ontología se les denomina *ontology learning* [1].

Los métodos de *ontology learning* pueden clasificarse en automáticos y semiautomáticos. Los automáticos son aquellos que no requieren de la intervención del usuario para la construcción, mientras que los semiautomáticos sí lo requieren [2]. De manera similar, se tienen métodos que buscan construir una ontología totalmente nueva, es decir, sin basarse en una construida anteriormente y métodos cuya finalidad es enriquecer o adaptar una ontología ya existente [1]. Para construir una ontología, se requiere primero extraer el conocimiento que la conformará, estas fuentes de conocimiento pueden ser diversas como textos, bases de conocimiento, diccionarios y otros [1]. Cuando el conocimiento se extrae a partir de un conjunto de documentos pertenecientes al dominio, se conoce como *ontology learning from text*.

Como primer paso para construir una ontología, se debe obtener una taxonomía. Una taxonomía es una clasificación de entidades de información a manera de jerarquías, de acuerdo a las relaciones de las entidades del mundo real que representan [3]. En una taxonomía solo se representan jerarquías entre conceptos, en cambio, en una ontología se puede representar diferentes tipos de relaciones, puede contener clases, subclases y reglas; en otras palabras, una taxonomía es una forma de representación más simple. Pese a no ser estructuras complejas como las ontologías, las taxonomías ayudan en la clasificación de la información, a realizar búsquedas de manera más eficaz y a navegar entre muchos conceptos [3][4][5]. De manera similar a las ontologías, la construcción de una taxonomía requiere mucho trabajo, pues se necesita extraer los conceptos más importantes a partir de un gran número de posibilidades y buscar sus respectivas relaciones de parentesco, por lo tanto, son necesarios métodos que permitan agilizar y facilitar su construcción.

En la construcción de una taxonomía, la extracción de los conceptos más relevantes o *key concepts* es un paso clave [2], pues éstas serán las entidades que conformarán la taxonomía [4]. Dado que los conceptos son usados como entradas principales en la construcción de la taxonomía, es necesario que estos sean relevantes en el dominio. Caso contrario, la calidad de la taxonomía desarrollada disminuirá pues clasificará conceptos poco importantes para el campo. Los conceptos clave, normalmente, son sustantivos o frases nominales [2], por lo tanto, se requieren aplicar técnicas de análisis de lenguaje natural, también conocidos como *NLP techniques*, para extraerlos. NLP o *Natural Language Processing* es una gama de técnicas computacionales para el análisis automático y representación del lenguaje humano [19].

Existen diversos métodos para extraer un conjunto de *key concepts* a partir de textos de un dominio específico como los presentados en text2Onto [7], en el cual se obtiene los conceptos clave según un análisis probabilístico, el resultado es asociado a la frase o palabra, con lo cual, se facilita el re-cálculo cuando se varían los documentos del *corpus* (conjunto de documentos sobre el que se trabaja); KP-Miner [8], el cual es un algoritmo construido en base a observaciones hechas sobre la extracción de frases clave, busca tanto frases como palabras que se repitan en los documentos, introduce un *boosting factor* el cual le da más peso a las frases más largas pues son menos propensas a repetirse que las palabras sueltas o frases

cortas; y KX [9], sistema de extracción de frases clave, este sistema usa análisis lingüísticos básicos y métodos estadísticos para la extracción, se extraen conjuntos de 2 a 4 palabras y se analizan según el número de veces que aparece en el documento y otros patrones elegidos por el usuario.

A inicios del 2014, se desarrolló un nuevo método para la extracción de conceptos clave conocido como CFinder, el cual combina *NLP techniques*, conocimientos estadísticos y conocimientos específicos del dominio para calcular la importancia de cada concepto y obtener aquellos que sean más relevantes en un dominio [2]. Este nuevo método ha probado ser más eficaz en la extracción de conceptos clave que KP-Miner y text2Onto [2]. Según los autores, el objetivo de este método es que sea usado para la construcción de ontologías u otro tipo de estructura que requiera una fase de extracción de conceptos clave. Sin embargo, en la actualidad, solo se ha incorporado en la extracción de conceptos clave.

En conclusión, se requieren construir taxonomías debido a las ventajas que ofrecen, sin embargo, requieren mucho esfuerzo para ser construidas a mano. Existen diversos métodos para la automatización, entre ellos los métodos que extraen el conocimiento de un grupo de textos. Para poder construir una taxonomía en base a un grupo de documentos, primero se debe extraer los conceptos más relevantes presentes en dichos textos. Luego, se debe deducir la jerarquía que, finalmente se convertirá, en la taxonomía. Para extraer los conceptos más relevantes, el método CFinder ha probado ser muy útil y superior a otros métodos, no obstante, no se ha integrado con un método que permita estructurar la jerarquía entre los conceptos extraídos. Por estos motivos, en este proyecto se buscará complementar el método CFinder con una fase para la deducción de jerarquías entre los conceptos extraídos y la construcción de una taxonomía, además de esta manera se estará brindando una nueva opción para la construcción automática de taxonomías.

1.2 Objetivo general

Implementar una herramienta que soporte la construcción de manera automática de una taxonomía en un dominio a partir de un conjunto de textos.

1.3 Objetivos específicos

Objetivo 1: Desarrollar un subsistema que permita la extracción automática de conceptos claves.

Objetivo 2: Desarrollar un subsistema que permita la identificación de jerarquías y construcción de una taxonomía de forma automática.

Objetivo 3: Comprobar el correcto funcionamiento de la herramienta realizando las pruebas necesarias a las taxonomías generadas.

1.4 Resultados esperados

En la tabla 1 se presentan los objetivos específicos y sus resultados esperados.

Objetivo 1: Desarrollar un subsistema que permita la extracción automática de conceptos claves.
• Resultado 1: Un componente de extracción de conceptos clave que pueda analizar los documentos del corpus y extraer una lista de candidatos. • Resultado 2: Un componente de extracción de conceptos clave que calcule el peso de cada candidato por cada documento. • Resultado 3: Un componente de extracción de conceptos clave que pueda identificar los conceptos clave de todo el corpus de documentos.
Objetivo 2: Desarrollar un subsistema que permita la identificación de jerarquías y construcción de una taxonomía de forma automática.
• Resultado 4: Un componente de identificación de jerarquías el cual sea capaz de obtener las relaciones de posibles padres para cada concepto. • Resultado 5: Un componente de identificación de jerarquías el cual sea capaz de construir una taxonomía en base a un corpus. • Resultado 6: Un aplicativo el cual integre el ingreso de datos, la fase de extracción de conceptos clave y la construcción de una taxonomía.
Objetivo 3: Comprobar el correcto funcionamiento de la herramienta realizando las pruebas necesarias a las taxonomías generadas.
• Resultado 7: El diseño de una prueba para probar las taxonomías generadas. • Resultado 8: Un grupo de documentos y una taxonomía de referencia para realizar las pruebas • Resultado 9: Un documento de pruebas donde se valide el correcto funcionamiento de la herramienta.

Tabla 1: Resultados esperados

1.5 Herramientas, métodos y procedimientos

En la siguiente sección se darán a conocer y se describirán las herramientas, métodos y procedimiento que se usarán durante el desarrollo del proyecto de fin de carrera para alcanzar los resultados esperados. En la tabla 2 se mapearán las herramientas, métodos y metodologías que se usarán para lograr cada resultado esperado y se explicará brevemente su uso.

Resultado esperado	Herramienta a usarse
RE1: Un módulo de extracción de conceptos clave que pueda analizar los documentos del corpus y extraer una lista de candidatos.	Se usará el método **CFinder [2]** para la obtención de conceptos clave, usando **GATE [19]** para el análisis de lenguaje natural y **NetBeans** [18] con **Java** para su depuración.
RE2: Un módulo de extracción de conceptos clave que calcule el peso de cada candidato por cada documento.	Se usará el método **CFinder [2]** para la obtención de conceptos clave, usando **GATE [19]** para el análisis de lenguaje natural y **NetBeans** [18] con **Java** para el cálculo de los pesos.
RE3: Un módulo de extracción de conceptos clave que pueda identificar los conceptos clave de todo el corpus de documentos.	Se usará el método **CFinder [2]** para la obtención de conceptos clave, usando **NetBeans**[18] con **Java** para la devolución de la lista ordenada.
RE4: Un identificador de jerarquías el cual sea capaz de obtener las relaciones de posibles padres para cada concepto	Se usará el IDE **NetBeans [18]**, el lenguaje **Java** para la integración de dichos módulos y se usará el **método *subsumption* [4]** para la obtención de los posibles padres de cada concepto
RE5: Un identificador de jerarquías el cual sea capaz de construir una taxonomía en base a un corpus.	Se usará el IDE **NetBeans [18]**, el lenguaje **Java** para la integración de dichos módulos y se usará el **método *subsumption* [4]** para la obtención de jerarquía y la construcción de la taxonomía
RE6: Una herramienta la cual integre el ingreso de datos, la fase de extracción de conceptos clave y la construcción de una taxonomía.	Se usará el IDE **NetBeans [18]**, la herramienta **GATE [19]** y el lenguaje **Java** para la integración de dichos módulos.
RE7: El diseño de una prueba para probar las taxonomías generadas.	Se usará el método **Golden estándar [4]** con la medida ***precision* [4] [14]** para determinar la calidad de las taxonomías construidas por la herramienta
RE8: Un grupo de documentos y una taxonomía de referencia para realizar las pruebas	Se usará el método **Golden estándar [4]** con la medida ***precision* [4] [14]** para determinar la calidad de las taxonomías construidas por la herramienta
RE9: Un documento de pruebas donde se valide el correcto funcionamiento de la herramienta.	Se usará el método **Golden estándar [4]** con la medida ***precision* [4] [14]** para determinar la calidad de las taxonomías construidas por la herramienta. Además se usará la herramienta

	Apache Jena para la lectura de la ontología que se tomará como muestra. [26]

Tabla 2: Mapa de herramienta y resultados

1.5.1 NetBeans

El IDE NetBeans fue creado en Java y además de ofrecernos un entorno de desarrollo en Java soporta múltiples lenguajes de programación tales como C, C++, PHP, HTML y otros. Es de software libre y multiplataforma gracias al lenguaje sobre el que fue implementado. Entre sus mayores atractivos está la inclusión de herramientas las cuales pueden generar la parte gráfica de distintas aplicaciones en poco tiempo [18]. En el proyecto se usará para el desarrollo de las partes donde sea requerido el uso del lenguaje Java y se usarán sus facilidades gráficas para la implementación de la herramienta.

1.5.2 GATE

GATE es una infraestructura la cual se usa para desarrollar componentes de software que procese lenguaje natural. Está desarrollada en java y es altamente extensible, cuenta con un gran número de *plug-ins* para distintas aplicaciones y un IDE para desarrollar nuevos componentes [19]. En el proyecto se usará GATE para desarrollar las herramientas que permitan extraer el contenido de los documentos que conforman el corpus. Se usará el IDE GATE Developer para construir una aplicación que analice los documentos y extraiga los candidatos a conceptos clave, posteriormente se usará GATE Embedded para el uso de dicha aplicación dentro de la herramienta objetivo del proyecto.

1.5.3 CFinder

CFinder es un nuevo método para la extracción de conceptos clave [2]. Este método será usado para la extracción de los conceptos con los cuales será construida la taxonomía. Entre sus características distintivas se puede mencionar que: i) es un método de extracción de conceptos clave no supervisado, es decir no requiere entrenamiento; ii) no requiere de corpus de múltiples dominios específico para la extracción de conceptos claves; iii) combina varias técnicas para la extracción de conceptos clave: técnicas de NLP, conocimiento estadístico, conocimiento específico del dominio y el patrón estructural de los términos extraídos; y iv) al incluir conocimiento específico del dominio y análisis del patrón estructural de los términos extraídos, CFinder evita los problemas a los que pueda llevar el uso único de la frecuencia como diferenciador de términos clave. Además, mediante el uso de técnicas de NLP y conocimiento estadístico, evita confiarse totalmente de los *input* del usuario.

Para la extracción de conceptos clave CFinder sigue tres pasos:

1. **Extracción de candidatos**: Se identifican en los textos todas aquellas palabras y frases que pueden ser conceptos clave mediante técnicas de NLP. Luego, se expanden las abreviaciones y eliminan los términos más comunes. Finalmente, se enriquece la lista de candidatos extrayendo frases derivadas de los candidatos ya formados.
2. **Cálculo del peso de los candidatos**: Se calcula el peso de cada uno de los candidatos por documento. Para esto se usa el conocimiento estadístico extraído del contenido del documento y conocimiento específico del dominio, que se obtiene de un glosario de conceptos específicos del dominio ingresado previamente.
3. **Extracción de conceptos clave**: Se suman los pesos de los candidatos obtenidos en todos los documento y se genera una lista ordenada con los conceptos clave.

El proceso seguido por CFinder se podría resumir de la siguiente manera:

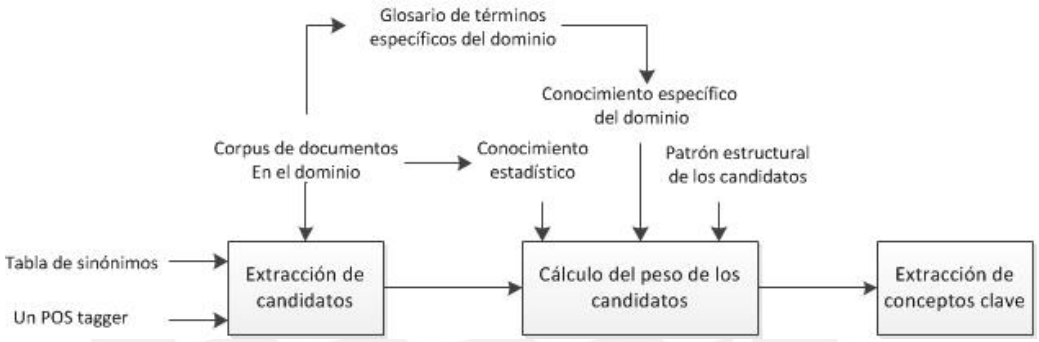

Imagen 1: Proceso del CFinder, extraído de [2]

Extracción de candidatos: El objetivo de esta fase es extraer todos los posibles candidatos a ser conceptos clave mediante técnicas de NLP. Para la extracción de los candidatos se requiere primero una fase de *POS tagging*. El proceso de *POS tagging* consiste en asignar a cada palabra del corpus una categoría léxica, tal como sustantivo, verbo, adjetivo, entre otros.

En esta etapa se sigue el siguiente proceso:

1. Se extrae los sintagmas nominales: Se buscan todos aquellos conjuntos de palabras compuesto por cero o más adjetivos seguidos de uno o más sustantivos.
2. Se buscan sinónimos y remueven los "*stopwords*": Se usa un diccionario de sinónimos para expandir los acrónimos a su forma extendida y se retiran los "*stopwords*", palabras comunes en el lenguaje, de las oraciones extraídas del corpus.
3. Enriquecimiento de candidatos: Además de los sintagmas nominales reconocidos, se consideran como candidatos los sustantivos extraídos como palabras sueltas y todas las combinaciones de palabras adyacentes

encontradas en el sintagma nominal que forme un sintagma nominal. A estos candidatos añadidos se les conoce como “frases dependientes”.

Cálculo del peso de los candidatos: Se realiza el cálculo de peso de los candidatos usando conocimiento estadístico y conocimiento específico del dominio. Para el conocimiento estadístico se usa la frecuencia de aparición del candidato a concepto clave, mientras más frecuente, mayor peso posee. Para el conocimiento específico se requiere el uso de un glosario de términos específicos del dominio, de esta manera es posible asignarle un mayor peso a dichos candidatos.

Para el cálculo del peso se distingue entre aquellos candidatos consistentes en solo una palabra y aquellos conformados por varias. Para aquellos candidatos conformados por una palabra, se realiza el cálculo de la siguiente manera:

$$w(c,d) = tf(c,d) * w_d(c),$$

$$tf(c,d) = \frac{f(c,d)}{max\ f(t,d)},$$

$$w_d(c) = 1 + \frac{log(df(c))}{log(max\ df(t))},$$

Donde *w(c,d)* es el peso del concepto *c* en el documento *d*, *tf(c,d)* representa el radio de frecuencia de *c* en *d* y es la parte del peso obtenida mediante conocimiento estadístico; *f(c,d)* es el número de apariciones de *c* en *d* y *max f(t,d)* representa el máximo número de apariciones de un candidato en el documento *d. w_d* es el peso obtenido por el conocimiento específico del dominio, *df(c)* es el número de veces que el término *c* aparece en el glosario y *max df(t)* es el máximo de veces que un término aparece en el glosario.

Para el cálculo del peso de un candidato conformado por muchas palabras, se realiza la suma de los pesos de los máximos *subsets* de candidatos que conforman dicho candidatos, donde un *subset* máximo es una frase que no es ningún *subset* de otro *subset* en el set de frases dependientes del candidato.

Extracción de los conceptos clave: Para cada concepto se realiza la suma de pesos en cada documento del corpus, luego se les ordena de forma descendente para obtener una lista ordenada de los conceptos. El usuario puede definir el número de conceptos clave que desea se le devuelvan.

1.5.4 Método Subsumption

El método Subsumption es usado para encontrar y establecer las relaciones de jerarquías entre conceptos usando la co-ocurrencia entre estos. En [4] es usada una variante para construir las relaciones de jerarquía entre conceptos y dar como resultado una taxonomía. La co-ocurrencia entre *x* e *y* se puede medir de la siguiente manera:

$$P(x|y) \geq t, P(y|x) < t$$

donde, *t* es una constante llamada trecho de co-ocurrencia. Esta ecuación significa que si *"x"* aparece en al menos la proporción *t* de todos los documentos donde *"y"* aparece e *"y"* aparece en una proporción menor a *t* de todos los documentos donde *"x"* aparece, entonces *"x"* es un *subsumer* de *"y".* En términos de una taxonomía, *"x"* sería el padre *"y".*

Dado que, según esta fórmula, un término podría tener varios padres, es necesario complementarla para poder hallar el mejor padre para cada término. Para esto se usan las siguientes fórmulas:

$$score(p, x) = P(p|x) + \sum_{a \epsilon A_p} w(a, x) * P(a|x)$$

donde, *p* es el padre potencial de *x*, A_p son todos los antecesores de *p* y *w(a,x)* es un peso calculado según la distancia, en niveles, entre el nodo *x* y el ancestro *a*.

$$w(a, x) = \frac{1}{d(a, x)}$$

donde , *d(a,x)* es la distancia en niveles entre los términos *a* y *x*.

1.5.5 Golden Estándar

Este método de evaluación se basa en comparar la taxonomía construida con una ya existente mediante diferentes medidas estadísticas. En este caso se usarán la medidas *precision* para comparar ambas taxonomías. *Precision, en el caso de los conceptos clave,* se puede definir como el radio de el número de conceptos clave identificados que se encuentran en la taxonomía versus el número de conceptos encontrados y, en el caso de las relaciones de subsunción, es el radio de relaciones de subsunción correctamente identificadas versus el número de relaciones de subsunción identificadas

1.5.6 MySQL

MySQL es un sistema *Open source* de administración de base de datos SQL desarrollada, distribuida y soportada por Oracle Corporation [25]. Durante el proyecto se usará para el manejo de la base de datos usada por la aplicación para el almacenamiento de conceptos, relaciones y relevancia calculados.

1.5.7 Apache Jena

Es una *framework* gratuito de java para la construcción de aplicaciones basadas en web semánticas y data interconectada. Durante el proyecto se usará para la búsqueda de conceptos en ontologías y de relaciones entre conceptos.

1.6 Alcance

El proyecto se encuentra dentro del área de las ciencias de la computación, específicamente en la ingeniería del conocimiento. El proyecto busca la implementación de una herramienta para la construcción automática de una taxonomía en base a un corpus de documentos en el lenguaje inglés dentro de un dominio específico del documento. Además, se usará el método CFinder para la extracción de los conceptos clave y se busca añadir a este método la fase de extracción de jerarquías y construcción de una taxonomía. Al momento de realizar la extracción de palabras clave, se requiere un glosario de términos específicos del dominio el cual será complicado encontrar para todos los dominios. El diseño de este glosario de términos no es parte de la solución. Los documentos usados como fuentes para la construcción de la taxonomía deberán ser de un mismo dominio, ya que la herramienta buscará armar solo una única taxonomía de todos los documentos.

1.7 Justificación

Se aportará al área del *ontology learning* un nuevo método y herramienta de extracción de una taxonomía en base a un conjunto de documentos. Se dará una nueva opción a todas aquellas personas con la necesidad de extraer conceptos y sus jerarquías de manera automática de un gran conjunto de documentos.

Los resultados obtenidos serán de gran ayuda a los investigadores que deseen usar el método de extracción de conceptos CFinder para futuras investigaciones relacionadas con la extracción y representación del conocimiento. Dado que el CFinder es un método recientemente desarrollado existe muy poca información de sus resultados al ser integrado en proyectos que busquen establecer relaciones entre los conceptos extraídos.

2 Marco Conceptual

En la siguiente sección se darán a conocer aquellos términos y conceptos que serán usados para entender tanto el problema objetivo como la solución planteada del presente proyecto. Se busca dar un mejor entendimiento de las taxonomías y sobre sus usos, además de las formas de extracción de conocimiento y de esta manera ayudar a comprender el porqué de una herramienta para la construcción automática de taxonomías en base a textos. Se presentarán los conceptos de ontología y taxonomía para poder entender el objetivo del problema planteado, además se darán distintas nociones de lo que es *ontology learning* y sus distintos tipos.

2.1 Ontología

Una ontología se puede definir como las palabras más comunes y conceptos usados para describir y representar un área del conocimiento. Se representan los objetos, propiedades, relaciones, funciones, reglas y restricciones entre ellos [3]. Normalmente las ontologías buscan conceptualizar el conocimiento de un dominio específico y su objetivo principal es el de proveer un entendimiento común y compartido de un dominio del conocimiento y promover la interoperabilidad entre las personas y muchos sistemas [2]. En pocas palabras busca representar el conocimiento de un dominio específico para que de forma que sea útil y pueda ser usado en diferentes contextos.

2.2 Taxonomía

Las taxonomías son estructuras similares a las ontologías. Pueden ser definidas como la clasificación de entidades de información a manera de jerarquías, de acuerdo a las relaciones de las entidades del mundo real que representan [3], es decir se representan solamente jerarquías. Las relaciones en una taxonomía son comúnmente “es una sub-clasificación de” o “es parte de” [3]. Una taxonomía puede ser representada como una estructura de árbol [3] [4], en el que cada hoja es llamada nodo. Cada nodo es un término que se está clasificando, donde los nodos hijos representan una subclase o parte del padre. Los usos básicos de una taxonomía son [3] [4] [5]:

1) **Clasificar**: Es similar a las taxonomías en la biología donde cada hijos es una subclase del padre.
2) **Buscar**: De manera similar a una búsqueda de libros en una biblioteca, se inicia por la raíz y se elige una sub-categoría cuyas características concuerden con lo que se busca, en esta sub-categoría existen otras con características más particulares, se continúa eligiendo aquellas cuyas características concuerden hasta llegar al término buscado.
3) **Navegar**: El ejemplo más claro se da en la web, donde se usan árboles de navegación. Los hijos son enlaces que aparecen en la página padre.

A diferencia de las ontologías, una taxonomía no busca representar todas las características del conocimiento, solo se representan las relaciones de jerarquía

entre los conceptos [3], pues no se representan relaciones con otros términos que no sean “hijos”, no existen propiedades o subclases.

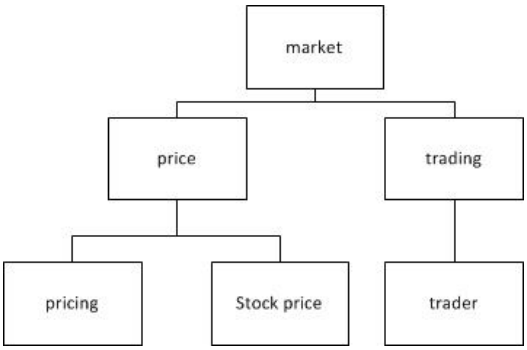

Imagen 2: Ejemplo de taxonomía, tomado de [4]

2.3 Ontology learning

Ontology learning se puede definir como un término genérico que agrupa todas las tareas de extracción de información y aquello métodos que se dedican a extraer automática o semi-automáticamente los conceptos y relaciones relevantes para ser incluido en una representación formal [6]. También se puede definir como el conjunto de métodos usados para disminuir los recursos requeridos para la construcción de una ontología [1]. Estos métodos se pueden subdividir en cinco tipos [1]:

- *Ontology learning from text*: El conocimiento para la construcción de la ontología es extraído de textos. Se usan técnicas de NLP para extraer los conceptos y relaciones.
- *Ontology learning from dictionary*: Se utiliza un diccionario que pueda ser leído por la máquina para identificar los conceptos y relaciones.
- *Ontology learning from a knowledge base*: Busca construir una ontología usando como fuente bases de conocimientos ya existentes.
- *Ontology learning from semi-structured data*: Propone construir una ontología obteniendo el conocimiento de fuentes que ya tienen datos estructurados.
- *Ontology learning from relation schemas*: Se extrae el conocimiento de bases de datos ya construidas para el desarrollo de la ontología.

2.4 Ontology learning from texts

Ontology learning from text consiste en extraer ontologías aplicando técnicas de análisis de lenguaje natural a los textos que contienen el conocimiento del dominio. Los métodos pertenecientes a *Ontology learning from texts* se pueden subdividir en [1]:

Extracción basada en patrones: Se reconoce las relaciones cuando se reconoce un patrón en una secuencia de palabras en el texto. Por ejemplo, si se encuentra un patrón “*auto* – ‘es un tipo de’ – *vehículo*”, se puede deducir que el término “*auto*” es

una instancia particular de "*vehiculo"*, es decir, en una jerarquía, *"vehiculo"* sería padre de *"auto"*.

***Association rules*:** Se busca la asociación entre los items contenidos en una base de datos. Se buscan los conjuntos *X* e *Y* tal que *X* e *Y* son conjuntos de items en las transacciones de una base de datos y que cada transacción que contenga a *X* tiende a contener a *Y*. Por ejemplo, si en una base de datos, se encuentran datos que se agrupan como una "*pre-venta*" y datos que se agrupan como "*persona*" y en una gran porción de transacciones donde toma lugar los datos de "*pre-venta*" también se usan los datos de "*persona*", se puede deducir que existe una relación entre "*pre-venta*" y "*persona*".

***Conceptual clustering*:** Los conceptos son agrupados según la distancia semántica entre ellos. Se busca construir jerarquías. Por ejemplo, si se tiene el término "*key concept*" y el término "*ontology learning*" en un documento y cada vez que el término "*key concept*" aparece en un párrafo también lo hace "*ontology learning*", pero no siempre que aparece "*ontology learning*" aparece "*key concept*", se puede deducir que "*ontology learning*" y "*key concept*" están relacionados siendo "*ontology learning*" de una jerarquía mayor a "*key concept*".

***Ontology pruning*:** Se busca construir una ontología para un dominio específico. Para esto, se requiere el uso de un diccionario que contenga términos importantes específicos del dominio y una ontología genérica ya construida. Primero, la ontología genérica es usada como base de la nueva ontología, luego el diccionario es usado para adquirir nuevos términos que serán integrados a la ontología base. Finalmente, el corpus de textos es usado para remover aquellos conceptos que no formaban parte del dominio específico.

***Concept learning*:** Se busca incrementar una taxonomía ya existente con conceptos extraídos de textos.

2.5 Key concepts

Los conceptos clave o *key concepts* son aquellas palabras o frases nominales las cuales se encuentran en el corpus de documentos y representan información útil e importante de un dominio específico [2] [6]. La razón de su importancia es que estos conceptos serán los nodos dentro de la taxonomía.

3 Revisión del estado del arte

En la siguiente sección se darán a conocer los resultados de la revisión de la literatura realizada para conocer el estado actual de la extracción automática de conceptos clave y la construcción automática de una taxonomía. Esto ayudará a conocer más sobre las distintas soluciones a los problemas y los criterios que se toman en cuenta para su construcción y validación.

3.1 Objetivos de la revisión

La siguiente revisión del estado del arte tiene los siguientes objetivos:

- Conocer más a fondo el tipo de análisis empleado en otras herramientas y métodos que identifican conceptos clave de documentos.
- Conocer el alcance de otros proyectos desarrollados, darse una idea del número de idiomas para los cuales son desarrollados y sus principales limitaciones.
- Obtener una idea aproximada del número de documentos usado para probar los métodos y herramientas en otros métodos
- Conocer los distintos métodos que fueron usados en otros proyectos para la obtención de las jerarquías entre conceptos.
- Averiguar el tipo de pruebas a las cuales se puede recurrir para validar una taxonomía generada de manera automática.

3.2 Método usado en la revisión del estado del arte

En esta revisión se usó las publicaciones contenidas en las bases de datos de SCOPUS e "IEEE". Estos artículos deberán incluir en su título o resumen las palabras "*key concept*", "*key concept extraction*", "*taxonomy*" y "*taxonomy extraction*", además, deberán pertenecer al área de Ciencias de la computación. Los artículos en la búsqueda no deberán de ser mayor a 5 años, sin embargo, se podrán incluir además aquellos proyectos que han sido referenciados en distintos artículos.

3.3 Estado del arte

3.3.1 Herramientas de *Ontology Learning* que usan *Key-concept extraction*

Text2Onto: Text2Onto es un *framework* para *ontology learning* de fuentes escritas [7]. Es un rediseño del sistema "*Text To Onto*" el cual se basa en la introducción de dos nuevos paradigmas:

- *Probabilistic Ontology Models* (POMs) el cual busca representar el resultado obtenido por el sistema junto con una probabilidad relacionada
- *Data-driven change discovery* el cual detecta los cambios en las fuentes de datos y calcula los cambios en la variación de los POMs relacionados a los cambios.

Estos paradigmas resuelven de por sí el problema del cambio en las fuentes de datos, pues no se necesita procesar nuevamente todos los documentos ni

reconstruir la ontología. Además ayuda a presentar más información al usuario del sistema al ayudarlo a seguir los cambios en la ontología según los cambios en las fuentes de datos. Gracias a las probabilidades relacionadas con las estructuras aprendidas, se puede presentar al usuario estas estructuras usando como filtro la certeza del sistema o como un ranking de estas.

Durante el proceso de extracción, se usó análisis de lenguaje natural para extraer tanto conceptos como relaciones. Para lograr dicho análisis, la herramienta elegida fue GATE por el set de herramientas con el que viene incluido y el hecho de ser fácilmente modificable de acuerdo a las necesidades. Finalmente se realiza un análisis estadístico.

KP-Miner: Es un algoritmo extractor de frases claves creado en el 2007 inicialmente para el idioma inglés y arábigo pero que puede configurarse para cualquier idioma [8]. Este extractor de frases clave no requiere de documentos de entrenamiento pues no usa técnicas de *machine learning*, sino que, usa el conocimiento del propio programador acerca del proceso de extracción de frases clave obteniendo un resultado igual o mejor al de extractores construidos en base a *machine learning*. El algoritmo KP-Miner posee tres pasos para la extracción de frases claves:

1. **Selección de candidatos:** Fase en la cual se extraen aquellas frases las cuales pueden ser frases clave. Una frase es reconocida tomando en cuenta que no debe estar separada por signos de puntuación y no debe incluir *stopswords*. Luego se aplican dos filtros, primero, la frase debe aparecer por lo menos “n” veces en el documento; segundo, se aplica una constante de *cutoff*, definida en número de palabras, según la cual, si una frase aparece por primera vez luego de dicha distancia del inicio del documento, no es tomada en cuenta.
2. **Cálculo de los pesos de los candidatos:** Fase en la cual se le asigna un peso a los candidatos elegidos. Dado que en los candidatos se pueden encontrar palabras sueltas y frases y estas últimas son menos frecuentes, se requiere usar un *Boosting factor* el cual equipara el peso obtenido entre las frases y las palabras.
3. **Refinamiento de la lista final de candidatos:** En este paso, se ordenan los candidatos de mayor a menor peso. Luego, en los n primeros candidatos, donde n es el número de candidatos a devolver, se busca si una frase es una subfrase de otro candidato, si es así, su cuenta es decrementada por la frecuencia del candidato del cual forma parte. Se recalculan los pesos y se vuelve a refinar. Finalmente, se retornan las frases clave.

KX: Es un sistema de extracción de frases clave desarrollado por FBK-IRST, este sistema usa análisis lingüísticos básicos y métodos estadísticos para la extracción [9]. Este sistema sigue los siguientes pasos para este propósito:

1. Se extrae del corpus una lista de *n-gramas*, donde *n-gramas* son cualquier conjunto de *n* palabras que se encuentran de forma seguida en el corpus. Además, cada *grama* es comparado con una lista de términos comunes, si el *grama* se encuentra en lista, entonces, no es tomado en cuenta.

2. Seleccionar una sublista de términos multi-palabra de la lista de *n-gramas* las cuales sean combinaciones de palabras que expresen un concepto. En KX, se extrae aquellos términos los cuales presentan un determinado patrón léxico, por ejemplo: artículo-sustantivo-adjetivo.
3. Para cada documento del corpus se reconocen y marcan los términos multi-palabra, luego, se aplica el *IDF* para todas las palabras y términos multi-palabras del corpus. Para esto, se usa un criterio basado en frecuencia. Se define el *Mincorpus* y el *Mindoc* donde *Mincorpus* es el número mínimo de veces que el *n-grama* debe aparecer en el corpus para ser considerado y el *Mindoc* es el mínimo número de apariciones en el documento actual. Finalmente para cada término y término *multi-palabra*, calcula su peso usando la fórmula log (Total Documentos/Documentos que contienen el término)
4. Este paso se considera sólo si es para un nuevo documento del cual se quiere extraer las frases clave. Primero se reconocen los términos multi-palabra usando el método del paso 3, luego, se cuenta la frecuencia de palabras y multi-palabras en el documento para obtener una lista ordenada de frases clave según su frecuencia.
5. Se re-procesa la lista obtenida según si los parámetros seleccionados
6. Finalmente, se busca en los términos de la lista acrónimos cuya forma extendida también está presente en la lista, de ser así, se suman las frecuencias y se elimina el acrónimo. También, se buscan entradas duplicadas, de encontrarse, se suman las frecuencias y se elimina una de ellas.

Sara tonelli et. al.: En [6] se presenta un ambiente colaborativo para la creación y expansión semi-automática de ontologías. Esta herramienta se construyó usando como base Moki [10] y fue enriquecido con un componente el cual permitiera la extracción de términos de recursos léxicos externos. Moki es una herramienta colaborativa basada en mediawiki, la idea principal de la herramienta es asociar una página wiki, que contenga información estructurada y no estructurada, a cada entidad de la ontología. Para la construcción o enriquecimiento de una ontología, esta herramienta sigue los siguientes pasos:

1. **Corpus Collection:** Los usuarios de la herramienta reúnen un corpus de un dominio específico para que sean la entrada del extractor de términos.
2. **Term extraction:** Se utiliza el extractor del algoritmo KX [9] para obtener los términos relevantes del corpus.
3. **Alignment with external resources:** Los términos obtenidos en el paso anterior son conectados con las entradas de WordNet, WordNet Domains y Wikipedia y le permite al usuario usar la información encontrada en dichas fuentes para agregarla a la ontología.
4. **Manual validation:** El usuario se encarga de decidir por último los términos y relaciones que serán ingresados en la ontología.

3.3.2 Métodos para la identificación de jerarquías de conceptos

M.Sanderson & B.Coft: Los autores proponen un método para derivar una organización jerárquica de los conceptos obtenidos de un grupo de documentos [13]. Se utiliza un tipo de co-ocurrencia conocida como subsunción (*subsumption*). Se utilizan cinco principios básicos:

- Los términos para la jerarquía son extraídos de los documentos y son los que mejor representan los temas tratados.
- La organización se dará de tal forma que el término padre será un término más general que el hijo, es decir el concepto padre subsume al hijo.
- El término hijo cubre un subtema del padre
- Se formará una jerarquía estricta donde cada hijo tendrá solo un padre
- Los términos ambiguos tendrán entradas separadas en la jerarquía

El método usado de la subsunción busca la cohesión entre dos términos y considera al más general como padre. Se considera "y" como hijo de "x" si, cada vez que el término "y" aparece, "x" aparece también, pero, si "x" aparece en un documento, "y" no aparece siempre.

Elias Zavitsanos et. al.: Similar al método de M.Sanderson y B.Coft [13], se propone la identificación de jerarquías mediante la subsunción entre los conceptos [14]. Sin embargo, a diferencia del anterior donde se realiza este proceso mediante probabilidad en una lista de conceptos, los autores en [14] proponen primero obtener una lista de sets de temas, mediante el modelo LDA [15]. Los términos de cada *set* de temas son evaluados por pares con los temas de sets diferentes buscando la independencia condicional. Esto se busca ya que, si existen dos temas relacionados A y B, pues pertenecen a un mismo set, y existe un tercero C que posibilita la independencia entre ambos, esto significa que los subtemas comunes de A y B son cerca a cero y C los contiene a ambos. Esta evaluación se realiza un algoritmo iterativo.

Woods: El método propuesto por Woods [17] también se basa en encontrar subsunciones entre los términos de un corpus, sin embargo, a diferencia de los anteriores, este propone hacerlo mediante un análisis léxico y no mediante un análisis estadístico. Se requiere de un *parser* que analice las frases que serán incluidas en la jerarquía, de esta forma se podrá obtener las estructuras sintácticas que componen las frases, por ejemplo:

"Car washing", Head noun: "washing", object: "car"
"Big red apple", Head noun: "apple", size: "red", size: "Big"

Posteriormente, se analizan las estructuras formadas, siendo los *Head nouns* términos más generales y los *head nouns* con modificadores términos más específicos. Se usan distintos axiomas como que "car" es una especificación de "vehicle" y "washing" una especificación de "cleaning".

Hearts: Este método propone la identificación de palabras en el idioma las cuales puedan servir para identificar relaciones de *hyponymy*[1] e *hypernymy*[2] como, en el idioma inglés: *"such as"*, *"kind of"* [16]. Posteriormente se realiza una búsqueda en el corpus de dichos patrones con lo cual se va obteniendo distintas jerarquías. El problema con este sistema cual como lo menciona su autor es que se requiere de intervención humana para identificar los patrones y si estos patrones significan relaciones de *hyponumy* o *hypernymy*.

3.4 Resumen

Proyecto	Tipo	Análisis usado	Alcance	Fuentes de datos usados para pruebas
Text2Onto	Automático	Análisis de patrones semánticos y análisis estadístico	Corpus de documentos en ingles	No se menciona
KP-Miner	Automático	Análisis estadístico	Corpus de documentos en inglés y árabe	7 corpus de documentos que contenían las frases clave ya anotadas por el autor de los documentos, en total sumaban 502 documentos
KX	Automático (requiere entrenamiento)	Análisis de patrones semánticos y análisis estadístico	Corpus de documentos en inglés	100 documentos de entrenamiento y 44 de prueba
Sara tonelli et. al.	Semi-automatico	Análisis de patrones semánticos y análisis estadístico	Corpus de documentos en inglés	390 documentos

Tabla 3: Resumen de herramientas con key-concept extraction

Proyecto	Análisis usado	Método de evaluación
M.Sanderson & B.Coft	Método subsumption (Variante 1)	Se usó una encuesta sobre un grupo de usuarios donde se le preguntaba que pensaba sobre un cierto número de relaciones encontradas
Elias Zavitsanos	Método subsumption (Variante 2)	Comparación entre la jerarquía identificada y otra ya construida
Woods	Análisis léxico	--
Hearts	Patrones de palabras	Se compara las relaciones obtenidas con aquellas contenidas en wordnet

Tabla 4: Resumen de métodos para la identificación de jerarquías

[1] A es un hyponym de B si A es un tipo de B
[2] A es un *hypernym* de B si B es un tipo e A

3.5 Conclusiones

Al extraer los conceptos clave es muy común desarrollar tanto un análisis de patrones como un análisis estadístico, se complementan. Además normalmente los proyectos se centran en un idioma específico aunque se trata de hacer la herramienta lo más adaptable posible. Para hacer las pruebas, no existe un número determinado de documentos para comprobar la validez, en algunos casos se usaron solo 44 y en otros 502 documentos. En cuanto a los métodos para obtener la jerarquía entre conceptos, existe una gran diversidad, cada herramienta usa una variante distinta. Finalmente, para validar una taxonomía, existen dos métodos básicos: La primera, es hacer una comparación entre la taxonomía construida y otra ya existente y, la segunda, validar la taxonomía construida mediante expertos en el campo.

4 Extracción automática de conceptos

Este capítulo tiene como finalidad describir los resultados alcanzados relacionados al objetivo 1, donde se busca desarrollar un módulo que permita la extracción de los conceptos más relevantes de u grupo de documentos en inglés y debe seguir el método CFinder. De la misma manera, se describirán, los pasos seguidos para su construcción. Para llegar a los resultados obtenidos, se ha utilizado el método CFinder en colaboración con las herramientas GATE y NetBeans.

4.1 Selección de candidatos

En este paso se busca implementar un componente de extracción de conceptos clave que pueda analizar un corpus de documentos en inglés y extraer una lista de candidatos que puedan ser considerados como conceptos clave. Este componente se desarrollará de tal manera que cumpla con los pasos establecidos por el método CFinder:

1) **Extracción de frases nominales**: Se extraen del texto todas aquellas palabras o frases las cuales estén conformadas por uno o más sustantivos o empiecen con uno o más adjetivos seguidos de uno o más sustantivos.
2) **Búsqueda de sinónimos y remoción de "*stopwords*":** Se usa un diccionario de sinónimos para expandir los acrónimos y abreviaturas, además se remueven aquellos candidatos que son conformados total o parcialmente por palabras comunes del idioma inglés.
3) **Enriquecimiento de candidatos**: Se obtienen nuevos candidatos a partir de aquellos conformados por más de una palabra. Se considera como nuevos candidatos a todas las combinaciones de palabras adyacentes dentro de un candidato que formen un sintagma nominal.

La siguiente prueba fue realizada sobre el documento *"CFinder An intelligent key concept finder from text for ontology development"* [2], el cual es un artículo académico en el idioma inglés que describe el método CFinder para la extracción de conceptos clave, el dominio de dicho documento es del *Ontology learning from text*. Este documento, cuya primera página puede observarse en la imagen 3, es de formato PDF y está conformado por 11 páginas que contienen 12566 palabras en total.

Expert Systems with Applications 41 (2014) 4494–4504

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

CFinder: An intelligent key concept finder from text for ontology development

Yong-Bin Kang, Pari Delir Haghighi, Frada Burstein *

Faculty of Information Technology, Monash University, 900 Dandenong Rd, Caulfield East 3145, Victoria, Australia

ARTICLE INFO

Keywords:
Key concept extraction
Keyphrase extraction
Domain-specific concept extraction
Ontology development
Ontology learning

ABSTRACT

Key concept extraction is a major step for ontology learning that aims to build an ontology by identifying relevant domain concepts and their semantic relationships from a text corpus. The success of ontology development using key concept extraction strongly relies on the degree of relevance of the key concepts identified. If the identified key concepts are not closely relevant to the domain, the constructed ontology will not be able to correctly and fully represent the domain knowledge. In this paper, we propose a novel method, named *CFinder*, for key concept extraction. Given a text corpus in the target domain, CFinder first extracts noun phrases using their linguistic patterns based on Part-Of-Speech (POS) tags as candidates for key concepts. To calculate the weights (or importance) of these candidates within the domain, CFinder combines their statistical knowledge and domain-specific knowledge indicating their relative importance within the domain. The calculated weights are further enhanced by considering an inner structural pattern of the candidates. The effectiveness of CFinder is evaluated with a recently developed ontology for the domain of 'emergency management for mass gatherings' against the state-of-the-art methods for key concept extraction including—Text2Onto, KP-Miner and Moki. The comparative evaluation results show that CFinder statistically significantly outperforms all the three methods in terms of F-measure and average precision.

1. Introduction

Due to an exponential growth of available information and knowledge, ontologies have become widely exploited in many different domains. Ontologies are typically built to formally conceptualize knowledge in a domain of interest. Their main aim is to provide a shared and common understanding of domain knowledge and promote interoperability between people and many including occasional axioms about the concepts from documents to build an ontology (Wong et al., 2012).

In an ontology, concepts typically represent a set of classes of entities or things within a domain (Noy & mcguinness, 2001). According to prior studies (Jiang & Tan, 2010; Li & Wu, 2006), concepts can be often described by *noun phrases* that are suitable for representing the key information within text documents. A noun phrase means a single noun or a group of words that function to-

Imagen 3: Primera página de [2]

Para la lista de *stopwords* se utilizaron las palabras encontradas en [21], las cuales son palabras comunes del lenguaje inglés. Por ejemplo:

```
aside
available
because
before
below
between
by
```

Además, se construyó un diccionario de sinónimos a partir de abreviaturas comunes identificadas en algunos documentos del dominio del *Ontology learning from text.* Para construir la lista se procedió a leer documentos del dominio, se identificó manualmente las abreviaturas y su forma extendida. Finalmente se obtuvo la siguiente lista de sinónimos:

```
POS: Part-Of-Speech
NLP: Natural language processing
```

```
WSD: Word sense disambiguation
ATCT: Automatic Taxonomy Construction from Text
TF: term frequency
IDF: inverse document frequency
TFIDF: term frequency - inverse document frequency
TF-IDF: term frequency - inverse document frequency
RTF: Relative Term Frequency
```

Al término del proceso, de las 12566 palabras del documento de prueba se obtuvieron 1656 candidatos. Algunos de estos candidatos obtenidos fueron:

```
key concept
extraction method
natural language processing
natural language
language processing
concept
```

Se pueden analizar los resultados obtenidos para corroborar que se cumplieron los pasos descritos anteriormente:

1) **Extracción de frases nominales**: Los candidatos extraídos del texto son conformados solo por uno o más sustantivos o están conformados por uno o más adjetivos seguidos de uno o más sustantivos. Por ejemplo:

```
key concept
extraction method
NLP
```

2) **Se buscan sinónimos y remueven los "*stopwords*":** En el documento se repetía numerosas veces el término “NLP” el cual se encuentra en la lista de sinónimos. Tal como se ve en los resultados, el término NLP fue sustituido por “*natural language processing*”. Además, como se puede visualizar, se han removido todos aquellos candidatos los cuales estaban conformados por alguna palabra encontrada en [21]. Al finalizar este paso, los candidatos son:

```
key concept
extraction method
natural language processing
```

3) **Se realiza un enriquecimiento de candidatos**: Luego de los pasos 1 y 2, en la lista de candidatos, no se observaban los términos “*natural language*” ni “*language processing*”, estos aparecieron en el enriquecimiento de candidatos. En este paso, se tomó al candidato “*natural language processing*” y se identificó todas las palaras adyacentes que cumplieran con la identificación del paso 1, es decir, que conformaran sintagmas nominales. Como resultado final, se obtuvieron los términos ya mencionados y fueron agregados también a la lista de candidatos. Finalmente, los candidatos finales son:

```
key concept
extraction method
natural language processing
natural language
language processing
concept
```

Construcción

Para la construcción del componente de extracción de conceptos clave se decidió separar cada una de las fases previamente mencionadas. Primero, se debe analizar el contenido del documento en busca de frases nominales, para ello se usó GATE Developer para la construcción de una aplicación "Candidatos" la cual realice esta labor. En la imagen 4, se puede observar el flujo de la construcción de dicha aplicación:

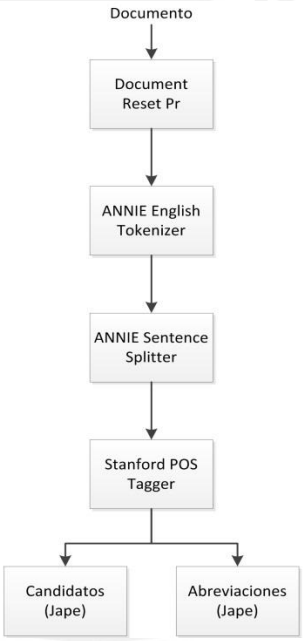

Imagen 4: Esquema de "Candidatos",
Imagen de autoría propia

El primer recurso, *Document Reset PR*, se encarga de remover todas aquellas anotaciones previas realizadas en el documento [23], luego *Annie English tokenizer* procesa el documento dividiendo el texto en *Tokens* los cuales poseen tipos como puntuación o palabra. Posteriormente, *Annie Sentence Splitter* se encarga de separar las distintas oraciones [23]. Una vez realizado esto, *Stanford POS Tagger* puede analizar cada oración y encontrar la categoría gramatical a la que pertenece cada *Token*, tales como sustantivo, verbo o adjetivo [23].

Finalmente, los documentos con las anotaciones realizadas ingresan a un par de recursos realizados en *Jape*, el primero, llamado "Candidatos" (Anexo A), se encarga de identificar las frases nominales y anotarlos como candidatos, para llevar

esto a cabo se programó el recurso de tal manera que extrajera aquel conjunto de palabras que cumpliera la expresión [(JJ|JJS)*(NN|NNS|NP|NPS|NNP|NNPS)+] donde “JJ” y “JJS” son los distintos tipos de adjetivos que se pueden encontrar, “NN”, “NNS”,”NP”,”NPS”,”NNP” y ”NNPS” son los diferentes tipos de sustantivos, “*” significa cero o más veces y “+” uno o más veces. En pocas palabras, se extraerán aquel conjunto de palabras conformados por cero o más adjetivos seguidos de uno o más sustantivos. Finalmente, como se puede ver en el anexo A, se usa la opción “*appelt*” en la cabecera del recurso, pues con esta opción se indica que siempre se elija la alternativa más larga encontrada en un rango si es que existen varias posibilidades. Por ejemplo, si se tiene *“key concept”*, existen 2 opciones, anotar “*key*” y anotar “*concept*” como candidatos o anotar *“key concept”*, al estar activada la opción “*appelt*”, se tomará *“key concept”* como la anotación correcta.

El segundo recurso se llama “Abreviaciones” (Anexo B), este se encarga de buscar todos aquellos *Tokens* o conjunto de *Tokens* que puedan ser considerados abreviaciones. Estas se obtienen identificando aquellas palabras que contengan dos o más combinaciones de una letra mayúscula y un punto, como “S.O.S.”, palabras conformadas solo con letras mayúsculas, como “POS” y palabras conformadas con letras mayúsculas y número como “Y2K”.

En la parte de la codificación en java, se realiza el siguiente proceso por documento: Primero, el documento es cargado y se usa la aplicación “Candidatos” construida en GATE para identificar tanto las frases nominales como las posibles abreviaciones en el contenido del texto. Luego, se pre-procesan las abreviaciones encontradas en el documento de tal manera que se tengan mapeadas solo aquellas que forman parte del diccionario de sinónimos. El código usado en la implementación del proceso descrito puede ser observado en el anexo C.

Una vez mapeada las abreviaciones, se obtienen las palabras anotadas como candidatos en el documento procesado. Estos candidatos pasan por un proceso de limpieza donde se retiran todos aquellos candidatos que contengan caracteres extraños en ellos. Además, los cambios de línea son borrados, se transforman múltiples espacios en blanco seguidos en uno solo y, si el candidato contiene algún *stopword*, es removido. Al finalizar la limpieza, se verifica si alguna de las abreviaciones se encuentra contenida en el candidato, si es así, se cambia por su forma extendida. Finalmente, como preparación para el paso de enriquecimiento, a cada uno de estos candidatos se le añade la cadena “ .\n” al final y son concatenados. El código usado en la implementación del proceso descrito puede ser observado en el anexo D.

Para el enriquecimiento de los candidatos, se creó la aplicación “Enriquecimiento” en GATE Developer, la cual se encarga de identificar todas las frases nominales que forman parte de cada candidato encontrado. Para esto, se usó el flujo mostrado en la imagen 4.

Esta aplicación es muy similar a “Candidatos”, debido a que, ambas deben encargarse de obtener frases nominales. Las pocas diferencias notables se encuentran en el recurso creado en *Jape* llamado “Enriquecimiento” (Anexo E).

Como puede observarse en el código fuente, en “Enriquecimiento” se usa la opción de control “*all*” en lugar de la opción “*appelt*” usada en “Candidatos”, esto permite que en “Enriquecimiento” se tomen todas las posibles frases nominales, no solo aquellas de mayor tamaño, usando el ejemplo anterior, si se tienen las opciones de anotar “*key*” y “concept” o *“key concept”*, se realizarán las tres anotaciones. Una vez concluido este proceso, se extraen los candidatos formados y se van almacenando en memoria, además, se guarda el número de veces que cada candidato está siendo extraído. El código usado en la implementación del proceso descrito puede ser observado en el anexo F.

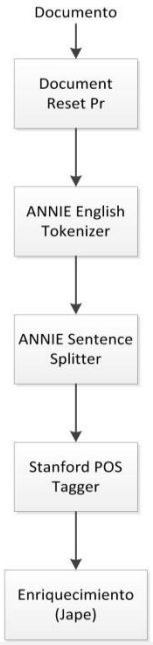

Imagen 5: Esquema de “Enriquecimiento”,
Imagen de autoría propia

Discusión

Para este primer resultado esperado, se realizó la implementación de un componente el cual puede extraer una lista de candidatos de cada documento perteneciente al corpus. Según las pruebas realizadas, el software implementado sigue los pasos establecidos por los autores del método y se obtiene como resultado final una lista de candidatos a conceptos clave. Estos candidatos tienen las características esperadas según lo descrito por los autores, por lo tanto, se puede concluir que se cumplió con el resultado esperado.

La obtención de los candidatos es la base para poder identificar los conceptos clave de un documento, los cuales, serán los componentes principales de la taxonomía armada. En pocas palabras, la extracción de estos candidatos es la base para todo el proceso de armado de la taxonomía.

Para realizar este componente se usó la metodología descrita por el método CFinder y se implementó en base a la descripción que dan sus autores, debido a

esto, solo se pueden identificar candidatos a conceptos clave de documentos en inglés. Sin embargo, es posible modificar este componente para que realice esta función en documentos de distintos idiomas.

4.2 Cálculo de pesos de los candidatos por documento

En este paso se busca implementar un componente el cual utilice los valores extraídos previamente para asignarle un peso a cada candidato por documento. Para establecer el peso de los candidatos se usarán los siguientes criterios:

Si el candidato está conformado por una palabra

$$w(c,d) = tf(c,d) * w_d(c),$$

$$tf(c,d) = \frac{f(c,d)}{max\ f(t,d)},$$

$$w_d(c) = 1 + \frac{log(df(c))}{log(max\ df(t))},$$

Donde *w(c,d)* es el peso del concepto c en el documento *d*, *f(c,d)* es el número de apariciones de c en d y *max f(t,d)* representa el máximo número de apariciones de un candidato en el documento *d. df(c)* es el número de veces que el término c aparece en el glosario de términos específicos del dominio y max df(t) es el máximo de veces que un término aparece en el glosario.

Si el candidato está conformado por más de una palabra, se realiza la suma de los pesos de los máximos *subsets* de candidatos que conforman dicho candidatos, donde un *subset* máximo es una frase nominal que no es ningún subset de otro subset en el set de frases dependientes del candidato.

Para la siguiente prueba se usó el documento [2] que describe el método CFinder para la extracción de conceptos clave, el dominio de dicho documento es del *Ontology learning from text*. A continuación, se presenta el diccionario de conceptos específicos del dominio usado en la prueba. Este diccionario fue construido en base al conocimiento que se tenía sobre el tema.

```
Ontology
Ontology Learning
Ontology Learning from text
Key concept
Taxonomy
Hierarchies
Taxonomy learning
Knowledge
```

Algunos resultados obtenidos del análisis del documento fueron:

```
Máximo número de apariciones de un candidato en el documento:
94
Máximo número de apariciones de una palabra en el glosario del
dominio: 3
```

Candidato	N° Apariciones en documento	N° Apariciones en glosario	Peso calculado
key concept extraction	28	--	2.3936170212765955
concept extraction	29	--	1.351063829787234
key concept	77	--	1.0425531914893615
Concept	87	1	0.925531914893617
Key	11	1	0.11702127659574468
Extraction	40	--	0.425531914893617
Ontology	50	3	1.0638297872340425

Tabla 5: Resultados del cálculo de pesos por candidatos

En el cuadro de resultados pueden observarse los pesos calculados de las palabras y los factores que intervinieron, tomando como ejemplo *ontology*, su peso sería calculado de la siguiente manera:

$$tf(ontology) = {}^{50}/_{94} = 0.531914894$$
$$wd(ontology) = 1 + {}^{\log(3)}/_{\log(3)} = 2$$
$$w(ontology) = tf(ontology) * wd(ontology) = 1.063829787$$

Por lo tanto, se cumple con el resultado cuando el candidato es conformado por una sola palabra. Si el candidato está conformado por más de una palabra, como el caso de *"key concept extraction",* se deben hallar la suma de sus máximos *subset*. En este caso, los máximos *subsets* son *"key concept"* y *"concept extraction"*, pero al ser este para también candidatos multi-palabra se debe hallar la suma de sus máximos *subsets*. El resultado se obtendría de la siguiente manera:

$$w(key\ concept\ extraction) = w(key\ concept) + w(concept\ extraction)$$
$$w(key\ concept) = w(key) + w(concept) = 1.0425531914893615$$
$$w(concept\ extraction) = w(key) + w(extraction) = 1.351063829787234$$
$$w(key\ concept\ extraction) = 2.3936170212765955$$

Construcción

Para este proceso, primero se calculan aquellos candidatos que son dependientes de otros, se busca por cada candidato conformado por dos o más palabras a otros candidatos que estén totalmente incluidos en él. Este proceso puede observarse en el Anexo G.

Luego se deben procesar aquellas palabras que se encuentran en el glosario de términos específicos del dominio. Para esto, se creó la aplicación *"Tokenizer"* en GATE Developer, la cual se usa para separar las palabras del documento. El esquema de la aplicación "*Tokenizer*" se puede observar en la imagen 6.

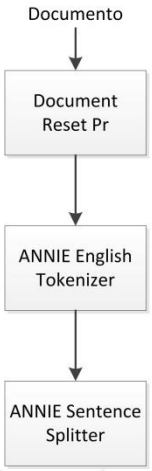

Imagen 6: Esquema de tokenizer,
Imagen de autoría propia

Una vez procesado el documento, se obtienen los "*Token*" y se va contando el número de veces que cada uno se repite. Posteriormente, se realiza el cálculo de pesos por candidatos en dos partes, en la primera, se obtienen los pesos de aquellos candidatos conformados por una sola palabra y en la segunda parte, de aquellos conformados por dos o más palabras. Para realizar esto de manera más efectiva, primero se ordenan los candidatos de menor a mayor según el número de palabras que poseen, luego por cada candidato de una palabra, se calcula los pesos "*tf*" y "*wd*" según los datos obtenidos y se multiplican para obtener el peso total del candidato. Luego, para calcular los pesos de los candidatos multi-palabra, se deben obtener los máximos *subsets* del candidato. Este proceso se realiza comparando entre sí los dependientes identificados previamente para obtener aquellos que no estén contenidos en su totalidad dentro de otro dependiente. A continuación, para obtener el peso del candidato, se suman los pesos individuales de cada *subset* máximo identificado. Es importante recalcar que sin el ordenamiento inicial, esto no sería posible, pues no se tendría la seguridad de que los pesos de los candidatos dependientes hayan sido calculados. De igual forma, se debe mencionar que al realizar las pruebas iniciales con los pesos calculados exactamente de esta manera, solían aparecer, con mucho peso, candidatos formados producto del azar, la mayoría de los cuales era por un error al leer tablas o diagramas. Para evitar este problema, se tomó la decisión de solo calcular los pesos de los candidatos multi-palabra que aparecieran por lo menos dos veces en el documento. El código fuente de los procesos descritos, pueden observarse en el anexo H.

Discusión

En este resultado esperado, se realizó la implementación de un componente que calcule el peso de cada candidato por documento del corpus. Según las pruebas realizadas, el software implementado sigue los pasos establecidos por los autores del método y da como resultado la lista de candidatos por documento con un peso asociado a cada uno, por lo tanto, se puede concluir que se cumplió este resultado esperado.

La obtención de los pesos por cada candidato es necesaria para obtener los conceptos más relevantes de cada documento y de un conjunto de documento. Estos pesos serán los que decidan si un candidato será un concepto clave y por lo tanto pasará a formar parte de la taxonomía.

Finalmente, es importante notar que uno de los factores en los que se basa la obtención del peso por cada candidato es el número de apariciones del candidato en el documento. Aunque funciona, nunca se depuran los candidatos según su significado, es decir, se pueden tomar dos candidatos que son sinónimos y puntuarlos como candidatos diferentes, cuando en realidad deberían ser considerados iguales.

4.3 Identificación de conceptos en el grupo de documentos

En este paso se busca implementar un componente de extracción de conceptos clave que obtenga el peso final de cada candidato en el grupo de documento y pueda devolver un número solicitados de estos ordenados de mayor a menor peso.

Para la siguiente prueba se usaron los documentos *"Deliverable 1.5: A survey of ontology learning methods and techniques"* [1], que trata sobre diferentes métodos y herramientas usadas en *ontology learning*, *"CFinder An intelligent key concept finder from text for ontology development"* [2] que trata sobre el método CFinder para la extracción de conceptos clave y *"A semantic approach for extracting domain taxonomies from text"* [4] en el cual se da a conocer un *framework* para la extracción de taxonomías en base a conceptos clave; todos temas tratados en los documentos pertenecen al dominio del *ontology learning.* Cabe mencionar que el documento [1] sobrepasa en tamaño en gran medida a los documentos [2] y [4] por lo tanto, se espera que los términos relevantes pertenecientes al documento [1] tengan un mayor peso que los términos relevantes de los otros documentos-. Se usarán los mismos parámetros descritos anteriormente para el cálculo de los pesos. Se pidió que se devolviera 10 conceptos clave algunos de los resultados se presentan en la tabla 6.

Concepto clave	Peso [1]	Peso[2]	Peso[4]	Total
domain ontology acquisition tool	2.4677	1.6382	2.4677	7.2619
ontology learning	3.1018	1.2765	0	4.3784
domain ontology	2.4677	1.6382	0	4.1060
Ontology	2	1.0638	0.2	3.2638

Tabla 6: Resultados de la obtención conceptos clave

Como se puede observar en la tabla 6, los pesos finales de cada concepto extraído son la suma total de los pesos calculados por documento. Igualmente se comprueba que dado la extensión del documento [1] los conceptos clave hallados tienen mayor influencia en el resultado final.

Construcción

Para este proceso, por cada documento, se extrae la lista de candidatos y sus pesos calculados. Si el candidato no ha sido procesado previamente, se le guarda junto con su peso en la lista de candidatos, caso contrario, se suma el peso del candidato al peso ya almacenado. Finalmente, se ordenan los candidatos por peso y se devuelven los N primeros, siendo N un número definido por el usuario. El código fuente de los procesos descritos, pueden observarse en el anexo I.

Discusión

En este resultado esperado, se realizó la implementación de un componente que calcule el peso de cada candidato en todo el corpus. Según las pruebas realizadas, el software implementado suma correctamente los pesos individuales de cada candidato por documento y obtiene el peso total del candidato en el corpus, por lo tanto, se puede concluir que se cumplió este resultado esperado.

Este paso obtiene el peso total de cada candidato en el corpus. Con este valor calculado, se pueden obtener aquellos candidatos de mayor peso los cuales serán considerados como conceptos clave y será en base a estos conceptos clave que se construirá la taxonomía del dominio de forma automática.

5 Construcción de una taxonomía

Este capítulo tiene como finalidad describir los resultados alcanzados relacionados al objetivo 2, donde se busca desarrollar un módulo que permita la construcción de la taxonomía del dominio en base a los conceptos clave identificados previamente. De la misma manera, se describirán, los pasos seguidos para su construcción. Para llegar a los resultados obtenidos, se ha utilizado el método *Subsumption* en colaboración con las herramientas NetBeans y MySQL.

5.1 Obtención de posibles padres por concepto

En este paso se busca implementar un identificador de jerarquías el cual sea capaz de obtener las relaciones de posibles padres de cada concepto con los otros. Este componente se desarrolló de tal manera que cumpliera con los pasos presentados por el método *Subsumption* presentado en [4]. Para hallar las relaciones de jerarquía se deben obtener todos los pares de conceptos *(x,y)* , donde "x" es un posible padre de "y" tal que:

$$P(x|y) \geq t, P(y|x) < t$$

Es decir que "x" aparezca en al menos el porcentaje *t* de los documentos donde "y" aparece y "y" aparezca en menos del porcentaje *t* de los documentos donde "x" aparece.

Prueba

La siguiente prueba se realizó sobre un conjunto de 257 *abstracts* de documentos de la base de datos "Scielo". Estos documentos tenían en común la palabra clave "*taxonomy*" y se usó un *becnhmark* de 0.7. Algunas de las posibles relaciones identificadas y retiradas son las siguientes:

Relación
```
Hijo: species nosferattus discus sp
Padre: sp
P(x|y): 1.0
P(y|x): 0.01
Acción: Aceptada como posible relación
```

Relación
```
Hijo: species nosferattus discus sp
Padre: species
P(x|y): 1.0
P(y|x): 0.006666666666666667
Acción: Aceptada como posible relación
```

Relación:
```
Hijo: species nosferattus discus sp
Padre: teeth
```

```
P(x|y): 1.0
P(y|x): 0.125
Acción: Aceptada como posible relación
```

Relación
Hijo: species nosferattus discus sp

```
Padre: type
P(x|y): 1.0
P(y|x): 0.045454545454545456
Acción: Aceptada como posible relación
```

Relación:

```
Hijo: species nosferattus discus sp
Padre: type species
P(x|y): 1.0
P(y|x): 0.125
Acción: Aceptada como posible relación
```

Relación

```
Hijo: species nosferattus discus sp
Padre: type species nosferattus discus sp
P(x|y): 1.0
P(y|x): 1.0
Acción: Retirada como posible relación
```

Relación

```
Hijo: amazon
Padre: analysis
P(x|y): 0.375
P(y|x): 0.25
Acción: Retirada como posible relación
```

Como es posible observar en la lista anterior, las relaciones cuyo P(x|y) es mayor o igual 0.7(*benchmark*) y su P(y|x) es menor a 0.7 son aceptadas como posibles relaciones. También es posible observar que aquellas relaciones donde no se cumple esta condición, son retiradas. Por lo tanto, se cumple con lo esperado pues se identifican de forma correcta las posibles relaciones según el método establecido.

Construcción

Para la construcción del componente, primero se obtiene de la base de datos todos los conceptos claves identificados previamente y el número de documentos en donde aparecen (Anexo J). Luego, se obtiene todos los pares de conceptos clave y

el número de documentos en los que ambos conceptos aparecen (Anexo K). Una vez obtenida esta información, se procede a verificar si se cumple la condición explicada previamente (Anexo L), de ser así, la relación obtenida es considerada como una relación de posible parentesco, caso contrario, es descartada.

Discusión

Para este resultado esperado, se construyó un componente el cual es capaz de identificar las relaciones de posible parentesco para cada concepto. Este proceso es importante para el resultado final pues se identificarán aquellas relaciones en donde un concepto puede ser el padre de otro y posteriormente, se elegirá la mejor opción, llevando a la construcción de una taxonomía. Según las pruebas realizadas, el software implementado identifica correctamente las posibles relaciones de parentesco, por lo tanto, se puede concluir que se cumplió este resultado esperado.

Es importante notar que el método de subsunción usado para la identificación de jerarquías, toma en cuenta la coocurrencia de términos en documentos completos y no toma en cuenta la cercanía de estos términos en el documento, por lo que es necesario de múltiples documentos para poder realizar el proceso de manera correcta. Se podría usar otra variante de este método el cual tome en cuenta la distancia entre apariciones de los conceptos clave y de esta manera no necesitar múltiples documentos.

5.2 Identificación de las jerarquías

En este paso se busca implementar un identificador de jerarquías el cual sea capaz de identificar la mejor relación de parentesco para cada término, para esto, cada una de las relaciones halladas previamente deben ser puntuadas y se debe seleccionar la mejor de todas. Para realizar puntuar cada relación de parentesco se deben usar las siguientes formulas:

$$score(p,x) = P(p|x) + \sum_{a \epsilon A_p} w(a,x) * P(a|x)$$

donde, *p* es el padre potencial de *x*, A_p son todos los antecesores de *p* y *w(a,x)* es un peso calculado según la distancia, en niveles, entre el nodo x y el ancestro a.

$$w(a,x) = \frac{1}{d(a,x)}$$

donde, *d(a,x)* es la distancia en niveles entre los términos *a* y *x*.

Prueba

Los siguientes resultados son parte de la prueba anterior, se realizó sobre un conjunto de 257 *abstracts* de documentos de la base de datos “Scielo”. Estos documentos tenían en común la palabra clave “*taxonomy*” y se usó un *becnhmark* de 0.7. En la imagen 7 se muestra una parte de la taxonomía resultante. Como se puede observar, se calculó que el mejor padre para “*species nosferattus discus sp*”

es “*type species”*. El cálculo puede ser entendido de la siguiente forma: Al calcular el padre de “*species nosferattus discus sp*” se tenía “*species*”, *“type species”*, entre otros, como posibles padres, por lo tanto se calcula el puntaje de esas relaciones. Dado que *species* no tiene padre el cálculo realizado fue directo:

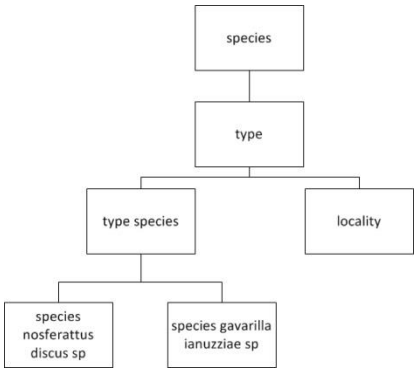

Imagen 7: Parte de la taxonomía generada,
Imagen de autoría propia

$$score(\text{“species”}, \text{“species nosferattus discus sp”}) = P(\text{“species”}, \text{“species nosferattus discus sp”}) + 0$$

$$\text{score}(\text{“species ”}, \text{“species nosferattus discus sp”}) = 1$$

En el caso de “*type species”,* tenía como padre a “*type*” y este a su vez a “*species*”, por lo tanto el cálculo del peso sería:

$$score(\text{“type species”}, \text{“species nosferattus discus sp”}) = P(\text{“type species”}, \text{“species nosferattus discus sp”}) + P(\text{“type”}, \text{“species nosferattus discus sp”}) \times \frac{1}{2} + P(\text{“species”}, \text{“species nosferattus discus sp”}) \times \frac{1}{3}$$

$$score(\text{“type species”}, \text{“species nosferattus discus sp”}) = 1 + 0.5 + 0.3333333333333333$$

$$score(\text{“type species”}, \text{“species nosferattus discus sp”}) = 1.8333333333333333$$

Dado que la relación final entre *“type species”* y *“species nosferattus discus sp”* es la de mayor puntaje se toma está como la relación de parentesco para *“species nosferattus discus sp”*.

Construcción

Para la construcción de este proceso, primero se crea un nodo raíz, el cual será padre de aquellos términos que no tengan un nodo padre al finalizar la asignación

de relaciones. Luego, cada nodo ingresa al procedimiento “identificaPadreNodo” el cual decidirá el mejor padre para el nodo.

El procedimiento “identificaPadreNodo”(Anexo M) sigue la siguiente lógica:

1) Si el nodo ya tiene un padre asignado, se termina el procedimiento
2) Si el nodo es el nodo raíz, se termina el procedimiento
3) Si el nodo no tiene posibles padres, se le asigna el nodo raíz como padre
4) En cualquier otro caso, se calcula, por cada posible relación de parentesco, un puntaje con la función “calculaScore” y se elige la de mayor puntaje.

Para la función “calculaScore” (Anexo N) la cual se encarga de obtener el puntaje de la relación de parentesco, se siguió la siguiente lógica:

1) Si el nodo ancestro es null, el puntaje es 0
2) Si el nodo ancestro es el nodo raíz, el puntaje es 0
3) Si el nodo ancestro no tiene un padre, se calcula el padre del nodo ancestro usando el procedimiento “identificaPadreNodo”
4) En cualquier otro caso, se calcula el puntaje de la relación entre el nodo hijo y el padre del nodo ancestro, luego, a este puntaje se le suma la probabilidad de ocurrencia del nodo ancestro si está presente el nodo hijo multiplicado por uno entre la distancia entre el nodo hijo y el nodo ancestro.

Es importante mencionar que uno de los parámetros de la función “calculaScore” es la distancia en niveles de los nodos involucrados en la relación, por ello, si se quiere calcular el puntaje entre un nodo y su padre, la distancia será 1, si es del nodo con su abuelo, la distancia será 2, etc.

Discusión

En este resultado esperado, se construyó un componente el cual es capaz de identificar la mejor relación de parentesco para cada concepto y construir una taxonomía en base a dichas relaciones. Este proceso es importante para el objetivo del proyecto pues se finalizará la construcción automática de la taxonomía. Según las pruebas realizadas, el software implementado calcula correctamente el puntaje de cada posible relación de parentesco y selecciona como la relación válida a aquella que mayor puntaje obtiene, por lo tanto, se puede concluir que se cumplió este resultado esperado.

Nuevamente, es importante notar que el método utilizado para puntuar las relaciones de posible parentesco toma en cuenta la coocurrencia de términos en documentos completos y no la cercanía de estos términos en el documento. Se podría usar otra variante de este método el cual tome en cuenta la distancia entre apariciones de los conceptos clave para el cálculo del puntaje de las relaciones.

5.3 Construcción de un aplicativo que integre el flujo completo

En este paso se busca implementar una aplicación mediante la cual se pueda realizar el ingreso de datos, la configuración del proceso y la visualización de la taxonomía extraída. Para la construcción de este aplicativo se usó sobre todo la

interfaz gráfica del IDE Netbeans. Primero, se construyó la interfaz de consulta desde la cual el usuario puede revisar los procesos de generación de taxonomía antiguos. Dicha consulta se puede observar en la imagen 8.

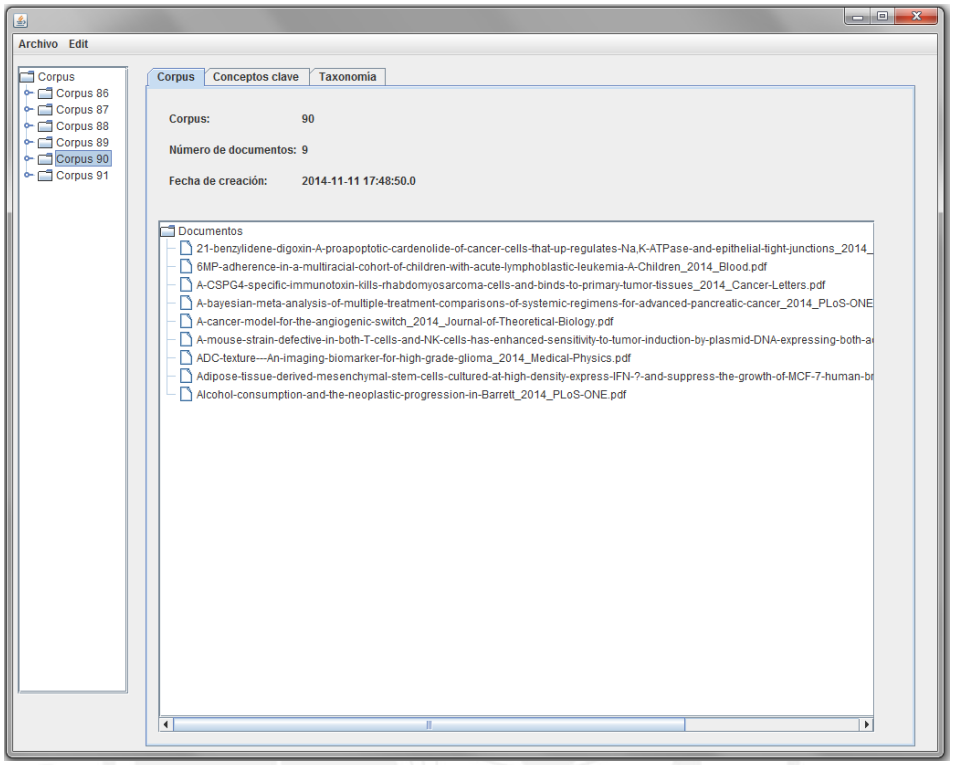

Imagen 8: Consulta de corpus,
Imagen de autoría propia

En la imagen 8, a la izquierda, se puede observar un árbol con todo el conjunto de corpus que se han creado hasta el momento. Al seleccionar uno, se carga la información de dicho corpus en las pestañas presentadas a la derecha. En la primera pestaña, “Corpus”, se puede observar la información básica del corpus, como su nombre, el número de documentos que lo conforman, la fecha de creación y los documentos incluidos en el corpus.

En la siguiente pestaña, “Conceptos clave” (Imagen 9) se pueden observar los conceptos clave identificados durante la ejecución del proceso. Además, se muestra el número de conceptos que se extrajeron y el mínimo número de apariciones de una frase para ser considerada como un candidato a concepto clave. Finalmente, en esta pestaña se tiene la opción de generar una nueva taxonomía en base a estos conceptos. Si es seleccionada esta opción, se abrirá una ventana en la que se puede configurar la variable “*benchmark*” y al seleccionar “Aceptar”, se dará inicio a la generación de una nueva taxonomía. Los resultados de las generaciones de taxonomías, podrán ser observadas desde la pestaña “Taxonomía”.

En esta pestaña (Imagen 10) se puede observar la taxonomía generada, la fecha de creación de la taxonomía y el benchmark usado para ser generada.

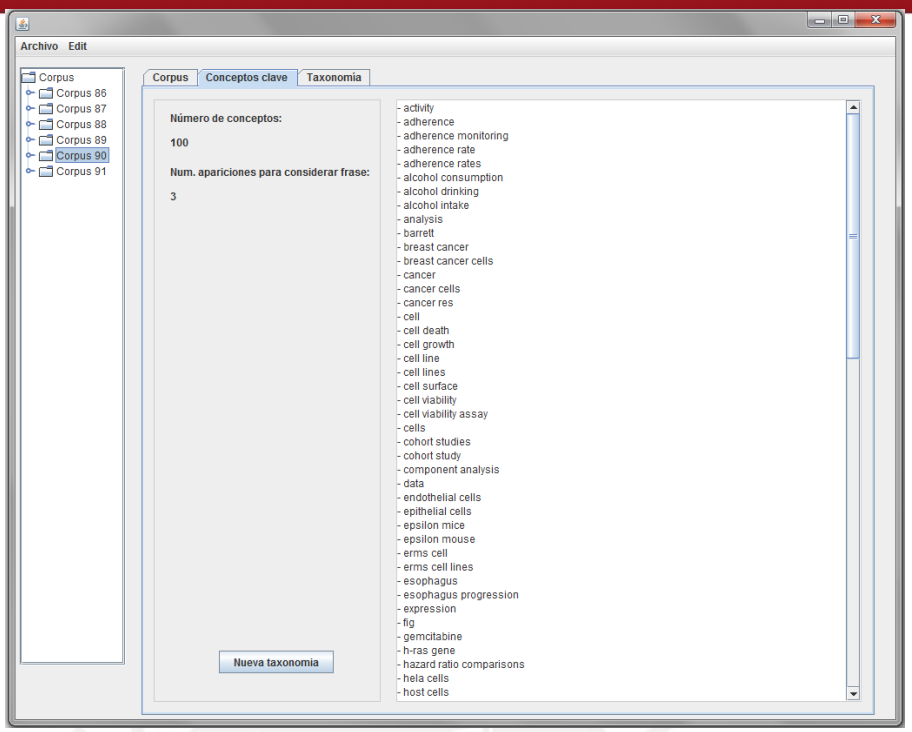

Imagen 9: Conceptos clave de un corpus,
Imagen de autoría propia

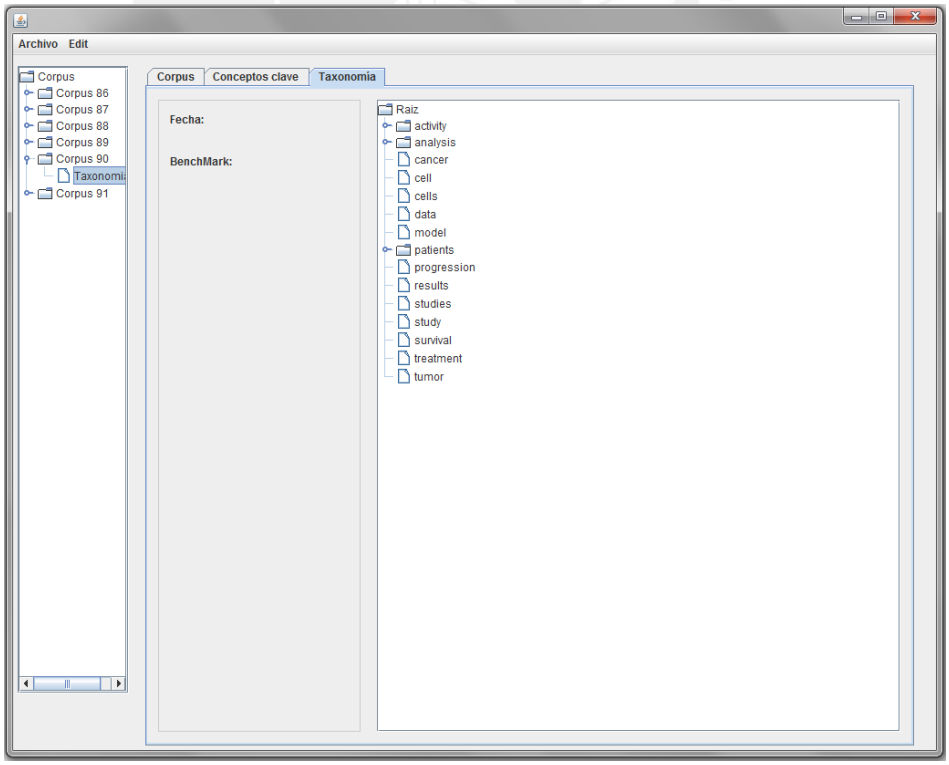

Imagen 10: Taxonomía de un corpus,
Imagen de autoría propia

Otra de las opciones disponibles es la de generar un nuevo corpus, para ello se debe seleccionar “Archivo>Nuevo corpus”, con lo cual aparecerá la ventana mostrada en la imagen 11. En esta ventana, se puede ingresar un nombre para el corpus, se puedo configurar el número de apariciones mínimas de una frase para ser considerada un candidato a concepto clave, el benchamark, el número de conceptos, se puede ingresar un glosario con conceptos resaltantes del dominio, el diccionario de abreviaturas y usando los botones “Agregar” y “Eliminar”, se puede editar los documentos que conformaran el corpus.

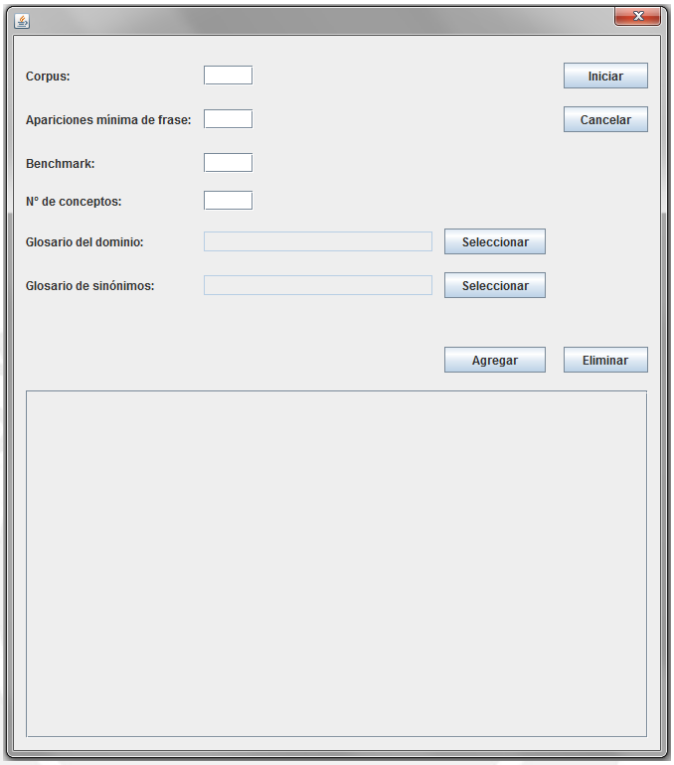

Imagen 11: Creación de un nuevo proceso,
Imagen de autoría propia

Finalmente se inicia el proceso y se generará un nuevo corpus y una taxonomía según los criterios ingresados.

6 Realización de las pruebas

Este capítulo tiene como finalidad describir el diseño de las pruebas realizadas y los resultados obtenidos de ellas. De la misma manera se interpretarán los resultados para determinar la eficiencia del método seguido. Para llegar a los resultados obtenidos, se ha utilizado el método Golden estándar en colaboración con las herramientas NetBeans y Jena.

6.1 Diseño de las pruebas a realizar sobre las taxonomías generadas

En este paso se busca diseñar las pruebas a seguir para comprobar el buen funcionamiento del método descrito. Para ello se seguirá el método Golden estándar el cual se basa en comparar la taxonomía obtenida mediante el método propuesto con otra ya construida y aceptada como correcta.

En estas pruebas se busca evaluar tanto la extracción de conceptos clave como la calidad de la taxonomía construida. Para ello se construirán taxonomías en base a 59, 88 y 113 documentos distintos de un mismo dominio y se comparará los conceptos clave y las relaciones halladas con los de una taxonomía ya existente del dominio. Posteriormente, se obtendrán la medida *presicion* para evaluar la efectividad del método.

En el caso de la obtención de conceptos clave, se extraerán 100 y 200 conceptos de cada set de documentos, además se dará como input de palabras relevantes del dominio un compilado de las palabras clave presentes en los documentos usados de prueba. Finalmente, se obtendrán la medida *presicion* de la siguiente manera:

$$presicion = \frac{N° \; conceptos \; obtenidos \; que \; aparecen \; en \; la \; taxonomia}{N° \; conceptos \; obtenidos \; en \; total}$$

En el caso de la taxonomía construida, se construirán distintas taxonomías en base al corpus cuya fase de obtención de conceptos clave posea la mayor *presicion.* Al construir la taxonomía se variará el valor del benchmark usado desde 0,05 hasta 0,95 en intervalos de 0,05 y se compararán los resultados obtenidos con cada uno. En este caso, se obtendrán la *presicion* de la siguiente manera:

$$presicion = \frac{N° \; relaciones \; obtenidas \; que \; aparecen \; en \; la \; taxonomia}{N° \; relaciones \; obtenidas \; en \; total}$$

6.2 Obtención de los inputs necesarios para las pruebas

En este paso se busca obtener tanto los documentos que serán usados para el desarrollo de las pruebas, así como, la taxonomía que servirá como referencia. Es importante recordar que tanto el grupo de documentos como la taxonomía, deben pertenecer a un mismo dominio.

Primero, se buscó una taxonomía de un dominio específico a la cual se pueda tener fácil acceso para realizar las comparaciones con la taxonomía generada. Debido a la estructura jerárquica en la que está construida, la facilidad para ser descargada y que trata sobre un tema en donde hay facilidad para encontrar artículos relacionados, se seleccionó la ontología *"National Cancer Institute Thesaurus".* Esta ontología es administrada por *"The National Center for Biomedical Ontology"* y tiene como dominio específico el cáncer.

Luego, se realizó la búsqueda de documentos relacionados al dominio de la taxonomía de referencia. Para ello se buscaron artículos publicados entre el 2012 al 2014 de la base de datos Scopus. Estos artículos debían ser del área de medicina y tener la palabra cáncer en su título o en sus palabras claves.

Finalmente para obtener los archivos con palabras clave del dominio, se incluyeron las palabras claves definidas por el autor del documento ya sea en el mismo documento, como en la base de datos Scopus. Además se revisó los documentos para obtener las abreviaciones e incluirlas en el diccionario de sinónimos.

6.3 Resultados

A continuación se mostrarán los resultados que se obtuvieron al realizar las pruebas respectivas y se les dará una interpretación de lo que significan para el proyecto. Primero se mostrarán los resultados obtenidos de la extracción de conceptos clave y luego de la taxonomía resultante.

Conceptos clave

Las siguientes tablas mostrarán la precisión obtenida en la extracción de 100 y 200 conceptos de distinta cantidad de documentos. Los conceptos extraídos serán considerados válidos si aparecen exactamente en la ontología de referencia.

	Corpus de 59 documentos	**Corpus de 88 documentos**	**Corpus de 113 documentos**	**Precisión del CFinder obtenida en [2]**
100 conceptos	0.46	0.5	0.54	0.525
200 conceptos	0.47	0.465	0.51	0.525

Tabla 7: Resultados de la prueba sobre la extracción de conceptos clave

Como se puede observar en la tabla 7, la precisión obtenida durante la extracción de conceptos clave, se encuentra alrededor del 50% de términos obtenidos usando la herramienta. En las pruebas realizadas por los autores del método CFinder, se obtuvo una precisión de 0.525, sin embargo, en estas pruebas también se consideraron aquellos conceptos que eran marcados manualmente como términos pertenecientes a la ontología, por lo que la siguiente tabla mostrará los resultados obtenidos con los términos exactos más aquellos considerados manualmente como válidos:

	Corpus de 59 documentos	Corpus de 88 documentos	Corpus de 113 documentos	Precisión del CFinder obtenida en [2]
100 conceptos	0.71	0.625	0.68	0.525
200 conceptos	0.625	0.69	0.65	0.525

Tabla 8: Resultados de la prueba sobre la extracción de conceptos clave considerando conceptos identificados manualmente como válidos

Construcción de una taxonomía

Para la prueba respectiva se formaron 19 taxonomías con los 100 conceptos extraídos de 113 documentos dado que esta fue la prueba que más precisión obtuvo en los conceptos extraídos. Al comparar las relaciones encontradas con aquellas relaciones en la taxonomía de referencia, no se encontraron relaciones que estuvieran tanto en la taxonomía extraída automáticamente como en la taxonomía de referencia. Esto da como resultado que en todos los casos de prueba se obtuviera una precisión de 0 respecto a la taxonomía de referencia. Sin embargo, pese a no tener precisión, se puede observar que cualitativamente, sí se logra agrupar términos relacionados de forma muy cercana.

Discusión

De los resultados obtenidos en las pruebas de extracción de conceptos clave, se puede determinar que el método CFinder posee una gran efectividad, pues un gran porcentaje de los conceptos extraídos formaban parte de la ontología de referencia, además, este porcentaje aumentó al considerar determinar de forma manual que algunos de los conceptos extraídos también formaban parte del grupo de conceptos relevantes. De igual manera se pudo identificar que algunos conceptos, aunque sintácticamente eran distintos, eran semánticamente iguales, pues eran sinónimos unos con otros o poseían una relación de pluralidad.

Los resultados obtenidos sobre la construcción de la taxonomía en base a los conceptos clave hallados demuestran que se logró agrupar conceptos similares en forma de jerarquías, sin embargo, no se logró construir una taxonomía que fuera precisa en relación a la taxonomía de referencia. La falta de precisión respecto a la taxonomía de referencia se puede deber a diversos factores, uno de ellos puede ser el número de documentos, dado que la identificación de jerarquías se basa en coocurrencias en un documento pero no toma en cuenta la cercanía de los conceptos dentro del mismo. Finalmente, otro factor a tomar en cuenta son los documentos seleccionados, los cuales pueden no haber sido los ideales para obtener toda la información requerida para el armado de la taxonomía. Otro de los factores importantes es que la taxonomía de referencia fue construida por seres humanos y por lo tanto, existe conocimiento implícito que ellos pueden haber aportado al armado de la ontología, así como, subjetividad en el orden y la clasificación de los elementos que la conforman.

7 Conclusiones

Para obtener el objetivo principal del proyecto se debían lograr tres objetivos específicos, en el primer objetivo se debía en construir un subsistema que realizará la extracción de conceptos clave, se utilizó para ello el método CFinder [2]. Para ello era necesaria la construcción de tres componentes: un componente de extracción de candidatos, un componente de cálculo de peso de cada candidato por documento y un componente que calcule el peso de cada candidato en todo el corpus [2]. En todos los casos, al realizar las pruebas, se obtuvo un resultado final que cumplía con las características debidas, por lo tanto, se concluye que se alcanzaron todos los resultados esperados para este objetivo y, por consiguiente, se consiguió el objetivo específico.

Para la obtención del segundo objetivo específico, según el método *subsumption* [4], era necesaria la construcción de dos componentes: un componente de extracción de candidatos, un componente de cálculo de peso de cada candidato por documento y un componente que calcule el peso de cada candidato en todo el corpus [4], además se requería construir un aplicativo que integrará la extracción de conceptos clave y la identificación de jerarquías entre ellos. Según las pruebas realizadas, el software implementado identifica correctamente las posibles relaciones de parentesco, calcula correctamente el puntaje de cada posible relación de parentesco y selecciona como la relación válida a aquella que mayor puntaje obtiene, por lo tanto, se puede concluir que se cumplió con los dos primeros resultados esperados. Por último, se construyó un aplicativo el cual era capaz de configurar e integrar la extracción de conceptos clave y la identificación de la construcción de una taxonomía, cumpliendo de esta manera el tercer resultado esperado para este objetivo. Al haberse cumplido con todos los resultados esperados, se puede concluir que se alcanzó el objetivo específico.

Finalmente, para la obtención del tercer objetivo específico, se requería realizar las pruebas sobre las taxonomías construidas para comprobar el funcionamiento de la herramienta desarrollada. Para esto se diseñaron las pruebas, se obtuvo los *inputs* necesarios y se realizaron las pruebas. En base a ellas se pudo demostrar la gran precisión de la extracción de conceptos clave y la diferencia entre las taxonomías generadas y la taxonomía de referencia. Al haberse realizado e interpretado el resultado de las pruebas, se puede afirmar que se alcanzaron todos los resultados esperados y se cumplió con el tercer objetivo específico.

Por lo previamente mencionado sobre los resultados del proyecto realizado se pueden obtener diversas conclusiones. En primer lugar, al haber conseguido alcanzar los tres objetivos específico se puede afirmar que se consiguió el objetivo principal del proyecto: La implementación de una herramienta que soporte la construcción de manera automática de una taxonomía de un dominio a partir de textos.

Además, los resultados confirman que el método CFinder, usado para la extracción de conceptos clave, posee una gran precisión al obtener conceptos relevantes en

un dominio específico pues, un alto porcentaje de los conceptos extraídos pertenecían a la ontología tomada como referencia. Sin embargo, se puedo observar que este método carece de un componente que le permita identificar conceptos según su significado, por lo tanto, es posible que se consideren términos que son equivalentes como conceptos distintos dentro de la lista de conceptos clave extraída.

Asimismo, aunque se logró armar una taxonomía de manera automática, esta no obtuvo una precisión satisfactoria al ser comparada con la taxonomía de referencia. Esto se puede deber a muchos factores como que el número de documentos usados no haya sido suficiente o que los documentos no eran los ideales para obtener las jerarquías de forma correcta. También se puede deber a que la taxonomía de referencia fue construida por seres humanos y por lo tanto aportan conocimientos implícitos, así como subjetividad en el orden y la clasificación de los elementos que la conforman.

7.1 Trabajos futuros

Para este proyecto se usó el método CFinder para la extracción de conceptos clave en el idioma inglés, dado que dicho método es de reciente publicación (Febrero 2014), solo ha sido trabajado en el idioma inglés. Por lo tanto, sería interesante adaptar el método a la gramática de distintos idiomas y usarlo en la fase de extracción de conceptos clave de documentos en otros proyectos. Además, en este método no se toma en cuenta el significado de los candidatos identificados, por lo que es posible identificar distintos conceptos clave con el mismo significado, por lo que, como trabajo futuro, se puede agregar al método CFinder una fase de depuración de candidatos según su significado.

Durante la fase de identificación de jerarquías se optó por usar una variante del método *subsumption* el cual toma la coocurrencia de conceptos en documentos completos, sin embargo, no se considera la distancia entre ellos para establecer si existe o no una relación. En un próximo proyecto se podría usar otra variante del método *subsumption* que tome en cuenta esta distancia u otro método completamente diferente para observar los resultados que se obtienen al integrarlos con el método CFinder.

8 Bibliografía

[1] Gómez-Pérez, Asunción, and David Manzano-Macho. *Deliverable 1.5: A survey of ontology learning methods and techniques* . Madrid.: Universidad Politécnica de Madrid, 2003.

[2] Yong-Bin , Kang, Pari Delir Haghighi, and Frada Burstein. *CFinder: An intelligent key concept finder from text for ontology development*. Expert Systems with Applications 41 (2014) 4494–4504.

[3] Daconta, Michael C., Leo Joseph Obrst, and Kevin T. Smith. *The Semantic Web a guide to the future of XML, Web services, and knowledge management*. Indianapolis: Wiley Pub., 2003. ISBN 0-471-43257-1.

[4] Meijer, Kevin, Flavius Frasincar, and Frederik Hogenboom. *A semantic approach for extracting domain taxonomies from text*. Decision Support Systems 62 (2014) 78–93.

[5] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American 284
(2001) 34–43

[6] Tonelli, Sara , Marco Rospocher, Emanuele Pianta, and Luciano Serafini Fondazione Bruno Kessler-irst. Boosting collaborative ontology building with key-concept extraction. Fifth IEEE International Conference on Semantic Computing, 2011.

[7] Cimiano, Philipp, and Johanna VÄolker. Text2Onto: A framework for ontology learning and data-driven change discovery: Institute AIFB, University of Karlsruhe, 2005.

[8] Samhaa R., El-Beltagy, and Rafea Ahmed. KP-Miner: A keyphrase extraction system for English and Arabic documents. Information Systems. 2007.

[9] Emanuele, Pianta, and Tonelli Sara. KX: A flexible system for Keyphrase eXtraction. Proceedings of the 5th International Workshop on Semantic Evaluation, ACL, 2010.

[10] Ghidini, Chiara, Marco Rospocher, and Luciano Serafini. MoKi: A Wiki-Based Conceptual Modeling Tool.

[11] Fortuna, Blaž, Marko Grobelnik, and Dunja Mladenić. Semi-automatic Datadriven Ontology Construction System: Department of Knowledge Technologies, Jozef Stefan Institute.

[12] Fortuna, Blaž, Marko Grobelnik, and Dunja Mladenić. Semiautomatic construction of topic ontology: Department of Knowledge Technologies, Jozef Stefan Institute.

[13] Sanderson, Mark, and Bruce Croft. Deriving concept hierarchies from text.

[14] Zavitsanos, Elias, Georgios Paliouras, George A. Vouros, and Sergios Petridis. Discovering Subsumption Hierarchies of Ontology Concepts from Text Corpora: IEEE/WIC/ACM International Conference on Web Intelligence, 2007.

[15] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation: Journal of Machine Learning Research 3, 2003.

[16] Hearst, Marti A.. Automated Discovery of WordNet Relations: To Appear in WordNet: An Electronic Lexical Database and Some of its Applications, Christiane Fellbaum (Ed.), MIT Press.

[17] Woods, William A.. Conceptual Indexing: A Better Way to Organize Knowledge. U.S.A.: Sun Microsystems, Inc. The SML Technical Report Series is published by Sun Microsystems Laboratories, a division of Sun Microsystems, Inc., 1997.

[18] "NetBeansIDE." *NetBeansIDE*. N.p., n.d. Web. 29 Julio 2014. <https://netbeans.org/>.

[19] Allemang, Dean, and James A. Hendler.Semantic Web for the working ontologist effective modeling in RDFS and OWL. 2nd ed. Waltham, MA: Morgan Kaufmann/Elsevier, 2011

[20] Cambria, Erik, and Bebo White. Jumping NLP Curves: A Review of Natural Language Processing Research. IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, 2014.

[21] "Full-Text Stopwords." MySQL. Oracle, n.d. Web. 30 Aug. 2014. <http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html>.

[22] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The Semantic Web." Scientific American May 2001.

[23] Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, et. al. Developing Language Processing Components with GATE Version 8 (a User Guide). The University of Sheffield, Department of Computer Science, 2014.

[24] A. Haslinda, and A. Sarinah. A Review of Knowledge Management Models. A: Journal of International Social Research, 2009.

[25] MySQL. N.p., n.d. Web. 20 Sept. 2014. <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>.

[26] Apache Jena. Web. 13 Nov. 2014.
<https://jena.apache.org/getting_started/index.html>.