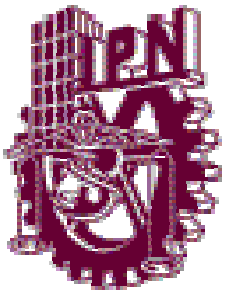


INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

LABORATORIO DE LENGUAJE NATURAL Y PROCESAMIENTO DE TEXTO

CONSTRUCCIÓN AUTOMÁTICA DE DICCIONARIOS SEMÁNTICOS USANDO LA SIMILITUD DISTRIBUCIONAL

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

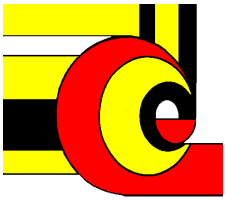
PRESENTA

L. I. MOISES EDUARDO LAVÍN VILLA

DIRECTORES DE TESIS:

DR. GRIGORI SIDOROV

DR. ALEXANDER GELBUKH



MÉXICO, D. F. **14 de Enero de 2010**

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 17:30 horas del día 16 del mes de Diciembre de 2009 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"CONSTRUCCIÓN AUTOMÁTICA DE DICCIONARIOS SEMÁNTICOS USANDO LA SIMILITUD DISTRIBUCIONAL"

LAVÍN	VILLA	MOISÉS EDUARDO
(Apellido Paterno)	(Apellido Materno)	(Nombre)

Con registro: B 0 7 1 4 5 1

aspirante al grado de: ***MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN***

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente
Dr. José Luis Oropeza Rodríguez

Secretario
Dr. Marco Antonio Moreno Ibarra

Primer vocal
(Director de tesis)
Dr. Grigori Sidorov

Segundo vocal
(Director de tesis)
Dr. Alexandre Felixovich Guelboukh Kahn

Tercer vocal
Dra. Sofia Natalia Galicia Haro

Suplente
Dr. Miguel Jesús Torres Ruiz

EL PRESIDENTE DEL COLEGIO

Dr. Jaime Álvarez Gallegos

INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACION
EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México D.F. el día 07 del mes Enero del año 2010, el que suscribe Moises Eduardo Lavín Villa alumno del Programa de Maestría en Ciencias de la Computación con número de registro B071459, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de Dr. Grigori Sidorov y DR. Alexander Guelbukh y cede los derechos del trabajo intitulado Construcción automática de diccionarios semánticos usando la similitud distribucional, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente correo electrónico eduarrrr@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Moises Eduardo Lavín Villa

Nombre y firma

Dedicatoria

A mis padres.

A mi hijo.

A mi familia.

A mis amigos.

AGRADECIMIENTOS

Quiero agradecer mis Asesores; el Dr. Grigori Sidorov y el Dr. Alexander Guelbukh por haberme dado la oportunidad de aprender de ellos.

ÍNDICE

Índice de Formulas

Índice de Tablas

RESUMEN

El desarrollo de ontologías, teniendo como entrada sólo documentos de texto es muy importante, ya que la construcción manual de ontologías es una labor extensa y costosa, sin embargo ya existe una gran cantidad de información en formato electrónico.

En esta tesis se presenta un método para la construcción automática de diccionarios semánticos, teniendo como entrada solamente documentos de textos, usando una serie de métodos estadísticos, entre ellos, la extracción de términos usando el algoritmo Log-Likelihood, la medida de similaridad coseno y el agrupamiento por medio del algoritmo K-Means.

ABSTRACT

The development of ontologies, taking as input only text documents is very important because the manual construction of ontologies is a labor extensive and expensive, however there is already a wealth of information in electronic format.

This thesis presents a method for automatic construction of semantic dictionaries, taking as input only text documents using a variety of statistical methods, including term extraction algorithm using the Log-Likelihood, the cosine similarity measure and clustering via K-Means algorithm.

1 INTRODUCCIÓN

1.1 Descripción del problema

La generación de diccionarios semánticos (ontologías) para un dominio, representa un desafío, ya que para generarlas se requiere de los servicios de uno o varios expertos en el dominio, con experiencia en el tema, y experiencia en la construcción de diccionarios semánticos (Uschold & Grunninger, 1996).

El método que se presenta en esta tesis, no requiere la intervención del usuario y es mucho más rápido que hacer un diccionario semántico manualmente.

1.2 Objetivos

1.2.1 Objetivo general

Crear un sistema que genere diccionarios semánticos automáticamente usando textos de un dominio en específico.

El sistema utilizará métodos estadísticos para la búsqueda de los conceptos y las relaciones entre ellos.

1.2.2 Objetivos específicos

Crear los módulos pertinentes al sistema automático de generación de diccionarios semánticos.

- Módulo para análisis de frecuencias de palabras.
- Módulo para extracción de conceptos.

    - Implementar método para cálculo de Log-Likelihood.
    - Implementar método para cálculo de Tf.Idf.
- Modulo para cálculo de Similaridad entre 2 palabras.
    - Implementar método del coseno del ángulo.
- Módulo para clusterización de términos.
    - Implementar el método K-Means.

Hacer los Siguientes Experimentos:

- Experimentos con un corpus de Informática, descargados de www.wikipedia.com, comparándolo con un corpus general que serán las noticias del periódico Excélsior.

Al terminar la tesis se esperan obtener los siguientes productos:

- Metodología de construcción de diccionarios semánticos a partir de los textos especializados, utilizando un conjunto de métodos estadísticos.
    - Extracción de términos. Utilizando el algoritmo Log-Likelihood.
    - Extracción de ciertas relaciones semánticas. Utilizando el método del coseno del ángulo.
    - Construcción del diccionario semántico.
- Prototipo de software que toma como entrada un conjunto de textos y construye un diccionario semántico tipo ontología en modo automático (sin alguna intervención del usuario).

1.3 Organización de la tesis

En el capítulo 2 se describe el estado del arte, se definen los conceptos a utilizar en esta tesis, se explican las herramientas existentes para la definición de diccionarios semánticos, se explican también los lenguajes para definir diccionarios semánticos y se hace una breve reseña de los trabajos actuales sobre la generación de diccionarios semánticos.

En el capítulo 3 se presenta la metodología utilizada para atacar el problema, se describe detalladamente los pasos a seguir.

En el capítulo 4 se detallan los experimentos realizados son el software, los resultados obtenidos, se hace un análisis de precisión y recall para tener una visión general de los resultados obtenidos.

2 ESTADO DEL ARTE

2.1 Conceptos básicos

Para poder entender los siguientes capítulos de esta tesis es necesario introducir los conceptos básicos con los que trabajaremos.

2.1.1 Stopword

Las las palabras denominadas stopwords son las que no incluyen información al dominio, para esta tesis, consideramos que las palabras que no son sustantivos, verbos, adjetivos o adverbios, son stopword. Estas palabras no serán tomadas en cuenta al construir el diccionario semántico.

2.1.2 Lematización

Las palabras con el mismo lema serán contabilizadas como una sola palabra (el lema), de modo que palabras como *trabaja*, *trabajar*, *trabajamos*, suman a la frecuencia de la palabra *trabajar* que es el lema de todas ellas.

2.1.3 Extracción de Términos Relevantes

Existen muchos modelos para ponderar términos en un corpus. El más conocido es Tf.Idf.

2.1.3.1 TF.IDF

Esta medida, permite comprobar el peso de un término (unidad léxica) en un documento (unidad de contexto).

2.1.3.1.1 Frecuencia de Términos

Se puede considerar para ponderar un término en un documento, la frecuencia absoluta ($\mathbf{tf_i}$) de éste, es decir, el número de veces que un término aparece en un documento. Un refinamiento de esta medida es considerar la frecuencia relativa en lugar de la frecuencia absoluta.

$$rtf_i = \frac{tf_i}{\Sigma_j tf_j}$$

Fórmula (1) Cálculo de la frecuencia relativa de un término

2.1.3.1.2 Frecuencia por Documento (DF_i)

A veces también es importante considerar el número de documentos ($\mathbf{df_i}$) donde aparece el término i, se puede escribir como:

$$[df]_i = |\{d \in D \mid d\ contiene\ i\}|$$

Fórmula (2) Cálculo de la Frecuencia por Documento

2.1.3.1.3 Inversa de la Frecuencia por Documento (IDF_i)

Esta medida penaliza a los términos que aparecen en muchos documentos. Su fórmula es la siguiente:

$$\mathbf{idf_i = \log_2 \frac{|D|}{df_i}}$$

Fórmula (3) Cálculo de la frecuencia relativa de un Término

Donde $\mathbf{|D|}$ es el total de documentos en el corpus.

2.1.3.1.4 TF.IDF

La frecuencia de términos (**tf**) y la inversa de la frecuencia en el documento (**idf**) se combinan para obtener una medida conocida como tf.idf.

$$tf.idf_{i,j} = tf_{i,j}\ x\ \log_2 \frac{|D|}{df_i}$$

Fórmula (4) Cálculo de Tf.Idf

Donde:

tf_{ij}= número de ocurrencias del término i en el documento j.

df_i = número de documentos que contienen i.

$|D|$ = número de documentos en una colección (corpus).

2.1.3.2 Log-Likelihood

Log-likelihood es una medida para comparar 2 corpus y ver que tan parecido es uno de otro (Rayson & Garside, 2000).

Para hacer el cálculo de Log-Likelihood, se toman en cuenta los valores observados (O) de cada palabra en los 2 corpus, el corpus de dominio y el corpus general, aplicando la fórmula 1 para calcular los valores esperados (E), es decir las ocurrencias que se esperaría que cada palabra tuviera en cada corpus.

$$E_i = \frac{N_i \sum_i O_i}{\sum_i N_i}$$

Fórmula (5) Cálculo de los Valores Esperados en un corpus.

Con estos valores observados y los valores esperados podemos calcular el Log-Likelihood con la siguiente fórmula:

$$-2\ln\lambda = 2\sum_i O_i \ln\left(\frac{O_i}{E_i}\right)$$

Fórmula (6) Cálculo de Log-Likelihood

Como en nuestro caso solamente usamos 2 corpus, uno de dominio y otro general, entonces el Log-Likelihood es calculado construyendo una tabla de contingencia como la siguiente:

	Corpus 1	Corpus 2	Total
Frecuencia de una Palabra	a	b	a+b
Frecuencia de otras palabras	c-a	d-b	c+d-a-b
Total	c	d	c+d

Tabla 2-1 Tabla de Contingencia para el cálculo de Log-Likelihood

Donde "c" corresponde al total de palabras en el corpus uno y "d" corresponde al total de palabras en el corpus dos (estos son los valores de N para la formula (1)). Los valores a y b serán los valores observados (O); considerando que necesitamos calcular los valores esperados (E), de acuerdo a la siguiente Fórmula (1), en nuestro caso N_1 = c y N_2 =d,

Entonces para una palabra cualquiera los valores esperados se calculan con las siguientes formulas:

$$E_1 = c * \frac{a+b}{c+d}$$ (Valores esperados para el corpus uno).

$$E_2 = d * \frac{a+b}{c+d}$$ (Valores esperados para el corpus dos).

Fórmula (7) Calculo de los Valores Esperados en 2 corpus

El cálculo de los valores esperados (E) toma en cuenta el tamaño de ambos corpus, por lo que no es necesario normalizar los datos antes de aplicar la fórmula.

Entonces podemos calcular el Log-Likelihood de acuerdo con la fórmula 2. Esto equivale a calcular el Log-Likelihood G_2 de la siguiente manera:

$$G_2 = 2 * \left(\left(a * \log\left(\frac{a}{E_1}\right)\right) + \left(b * \log\left(\frac{b}{E_2}\right)\right)\right)$$

Fórmula (8) Calculo de Log-Likelihood para 2 corpus

Veamos un ejemplo:

Supongamos que tenemos 2 corpus, El corpus uno (C1) con 1000 palabras y otro (C2) con 20000 palabras. En estos corpus tenemos las palabras A, B, C, D, E y F entre otras, con los valores observados que se muestran en la siguiente tabla de contingencia:

Palabra	Corpus 1 (C1)	Corpus 2(C2)	Total
A	5	50	55
B	5	200	205
C	5	500	505
D	10	50	60
E	10	200	210
F	10	500	510
Total	1000	20000	21000

Tabla 2-2 Tabla de Contingencia de ejemplo

Calculamos los valores esperados para cada palabra con las formulas E_1 y E_2 de la fórmula (2-7) en la siguiente tabla:

Palabra	C1(O_1)	C2 (O_2)	Total	E_1	E_2
A	5	50	55	2.619048	52.380952
B	5	200	205	9.761905	195.238095
C	5	500	505	24.047619	480.952381
D	10	50	60	2.857143	57.142857

Palabra	C1(O_1)	C2 (O_2)	Total	E_1	E_2
E	10	200	210	10.000000	200.000000
F	10	500	510	24.285714	485.714286
Total	1000	20000	21000		

Tabla 2-3 Cálculo de Valores Esperados

Nótense los siguientes hechos entre los valores observados y esperados:

- El valor O_1 puede ser menor que el valor E_1 y el valor O_2 mayor que E_2. (palabras A, C y F).
- El valor O_1 puede ser igual que el valor E_1 y el valor O_2 igual que E_2. (palabra E).
- El valor O_1 puede ser mayor que el valor E_1 y el valor O_2 menor que E_2. (palabras B y D).

Ahora calculamos el Log-Likelihood con la formula (2-8).

Palabra	O1	O2	Total	E1	E2	Log-Likelihood
A	5	50	55	2.619048	52.380952	1.814270
B	5	200	205	9.761905	195.238095	2.948524
C	5	500	505	24.047619	480.952381	23.133853
D	10	50	60	2.857143	57.142857	11.702120
E	10	200	210	10.000000	200.000000	0.000000
F	10	500	510	24.285714	485.714286	11.241473
Total	1000	20000	21000			

Tabla 2-4 Cálculo de Log-Likelihood

Nótense los siguientes hechos:

- Los valores de log-likelihood siempre son valores positivos.
- Las palabras F y D tienen un valor de Log-Likelihood similar a pesar de que sus valores observados y esperados son muy distintos.

La palabra D debería de tener un Log-Likelihood superior a la palabra F y la palabra E debería de tener un Log-Likelihood superior a la palabra F; por lo tanto la palabra F debería de tener un valor negativo.

- Para hacer este ajuste, debemos de hacer el cálculo de las frecuencias relativas, es decir, aplicar la fórmula (2-1):

	Corpus 1	Corpus 2	Total	FR1	FR2
Frecuencia de una Palabra	a	b	a+b	(a/c) * 100	(b/d) * 100
Frecuencia de otras palabras	c-a	d-b	c+d-a-b		
Total	c	d	c+d		

Tabla 2-5 Ajuste al cálculo de Log-Likelihood

Si la frecuencia relativa es mayor en el corpus 2, indica un mayor porcentaje de uso en el corpus 2, es decir, es menos relevante en el corpus de dominio, en este caso, el valor se multiplica por (-1) para que no aparezca como relevante en el dominio.

2.1.4 Medidas de Similaridad

La medida de similaridad se define usualmente por proximidad en un espacio multidimensional (Cimiano, 2006).

La formula de las medidas más utilizadas son:

2.1.4.1 Medidas de similaridad Binarias

Las medidas más conocidas para asignar similaridad entre vectores binarios, por ejemplo, conteniendo sólo valores entre 0 y 1, son el coeficiente de DICE y el coeficiente de Jaccard.

2.1.4.1.1 Coeficiente de DICE

$$Dice(x,y) = \frac{2\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Fórmula (9) coeficiente de DICE

2.1.4.1.2 Coeficiente de Jaccard

$$Jaccard(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i y_i}$$

Fórmula (10) Coeficiente de Jaccard

2.1.4.2 Medidas de similaridad Geométricas

La medida de similaridad geométrica que utilizaremos en esta tesis es el *coseno del ángulo* entre 2 vectores, es del rango de 1 para vectores que apuntan en la misma dirección, hasta -1 para vectores que apuntan en la dirección opuesta, en aplicaciones de minería de textos, donde las dimensiones de los vectores corresponden a conteos de frecuencias de palabras, los vectores nunca apuntan en dirección opuesta.

Las medidas de similaridad geométricas más conocidas son:

2.1.4.2.1 Distancia Euclidiana:

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

Fórmula (11) Distancia Euclidiana

2.1.4.2.2 Distancia Manhattan:

$$L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Fórmula (12) Distancia Manhattan

2.1.4.2.3 Distancia Minkowski:

$$L_q(x,y) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q}$$

Fórmula (13) Distancia Minkowski

Donde, si q = 1 estamos hablando de la distancia Manhattan y si q = 2 entonces estamos hablando de la distancia Euclidiana.

2.1.4.2.4 Similaridad por Coseno

$$\cos(x,y) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Fórmula (14) Distancia por coseno

2.1.5 Clustering

El Agrupamiento, o Clustering, es el proceso de agrupar datos en clases (clústeres) de manera que los elementos de un clúster tengan una similaridad alta entre ellos, y baja similaridad con objetos de otros clústeres.

La medida de similaridad está basada en los atributos que describen a los objetos. En nuestro caso, la medida de similaridad está basada en las ocurrencias de cada término en los diferentes documentos del corpus de dominio.

Existen muchos algoritmos de clustering, he aquí dos de ellos:

- Métodos basados en particiones: En estos métodos se construyen k particiones del conjunto de datos, donde cada partición representa un grupo o clúster. Cada grupo tiene como mínimo un elemento (centro) y cada elemento pertenece a un solo clúster. Estos métodos, crean una partición inicial e iteran hasta un criterio de paro. Los más populares son k-means y k-medoids.

- Métodos jerárquicos: crean descomposiciones jerárquicas. Y existen de 2 tipos:

    - El método aglomerativo o bottom-up, empieza con un grupo por cada objeto y une los grupos más parecidos hasta llegar a un solo grupo u otro criterio de paro.

    - El método divisorio o top-down, empieza con un solo grupo y lo divide en grupos más pequeños hasta llegar a grupos de un solo elemento u otro criterio de paro.

Para los fines de esta tesis elegimos el algoritmo k-means que es del tipo de algoritmos de agrupamiento basado en particiones.

2.1.5.1 K-means

Su funcionamiento es el siguiente (MacQueen, 1967):

- Toma como parámetro k que es el número de clústeres que forma.

- Selecciona k elementos aleatoriamente, los cuales representan el centro o media de cada clúster. A cada objeto restante se asigna al clúster con el cual más se parece, basándose en una medida de similaridad entre el objeto y la media del clúster.

- Después se calcula el nuevo centro del clúster y se itera hasta no cambiar de centro.

Selecciona k objetos aleatoriamente **Repetir** Reasigna cada elemento al clúster más similar Actualiza el valor de las medias de los clústeres **Hasta** no hay cambio

Tabla 2-6 Algoritmo K-Means

2.2 Similitud distribucional

En la similitud distribucional una palabra puede ser representada por un **vector de coocurrencia**, en donde cada entrada corresponde a otra palabra. El valor de una entrada especifica la frecuencia de la ocurrencia de las dos palabras en el corpus, es decir, la frecuencia en la cual ellas coocurren con alguna relación particular en el texto. El grado de similaridad entre un par de palabras es computado por alguna similaridad o medida de distancia en que es aplicada al correspondiente par de vectores. Tomaremos esta similaridad para crear una ontología.

2.3 Definición de ontología

En la documentación consultada para esta tesis la definición más utilizada es la de Gruber, quien en su definición corta de ontología dice que una ontología es una especificación explicita de una conceptualización (Gruber, 1993). Donde una conceptualización es una abstracción, una visión simplificada del mundo que deseamos representar para algún propósito. Esto es; una ontología es una descripción de los conceptos y relaciones que pueden existir entre un agente o una comunidad de agentes.

En wikipedia. Se define el término **ontología** (en informática) como a la formulación de un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la

finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas.

Una ontología define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores. Se usa la palabra "dominio" para denotar un área específica de interés o un área de conocimiento.

Las ontologías consisten de términos, sus definiciones y axiomas que los relacionan con otros términos que están organizados en una taxonomía. Estos axiomas permiten realizar búsquedas con inferencias. Las ontologías definen conceptos y relaciones de algún dominio, de forma compartida y consensuada; y esta conceptualización debe ser representada de una manera formal, legible y utilizable por las computadoras.

Para El Consorcio World Wide Web (W3C), una ontología define los términos que se usan para describir y representar un cierto dominio. Uso la palabra "dominio" para denotar un área específica de interés o un área de conocimiento.

Las ontologías que desarrollaremos en esta tesis abarcan las palabras clave de un dominio dado, y un grado de similaridad entre ellas.

2.3.1 Razones para desarrollar diccionarios semánticos

Hay varias razones para desarrollar diccionarios semánticos.

- Para tener una mejor comprensión de algún área del conocimiento.
- Para que otros comprendan algún área del conocimiento.
- Para hacer consensos en la interpretación del conocimiento.
- Para permitir que las máquinas utilicen el conocimiento en alguna aplicación.

- Para permitir el intercambio de conocimiento entre las máquinas.

2.3.2 Usos de los diccionarios semánticos

Los diccionarios semánticos tienen diferentes usos, entre ellos podemos mencionar los siguientes:

- Favorecer la comunicación entre personas, organizaciones y aplicaciones, porque proporcionan una comprensión común de un dominio, de modo que se eliminan confusiones conceptuales y terminológicas.

- Lograr la interoperabilidad entre sistemas informáticos.

- Razonar automáticamente. Partiendo de unas reglas de inferencia, un motor de razonamiento puede usar los datos de los diccionarios semánticos para inferir conclusiones de ellos.

- Entender cómo diferentes sistemas comparten información.

- Descubrir ciertas distorsiones presentes en los procesos de aprendizaje en un mismo contexto.

- Formar patrones para el desarrollo de Sistemas de Información. Por lo tanto, el uso de diccionarios semánticos en el desarrollo de Sistemas de Información permite establecer correspondencia y relaciones entre los diferentes dominios de entidades de información.

2.3.3 Objetivos de los diccionarios semánticos

Se puede decir de los diccionarios semánticos que tienen los siguientes objetivos principales:

Compartir la comprensión común de la estructura de información entre personas o agentes de software, lo que debe revertir de forma positiva y casi necesaria en la

extracción y recuperación de información, en páginas web, de contenidos conectados temáticamente.

Permitir la reutilización del conocimiento perteneciente a un dominio. Por ejemplo, a la hora de iniciar la elaboración de una ontología.

Permite hacer explícitos los supuestos de un dominio. Esta aseveración puede conducir a conclusiones muy interesantes para la representación del conocimiento más allá de consideraciones técnicas, operativas e informáticas.

Separa el conocimiento de un dominio del conocimiento que se puede denominar operacional. Con esto se alude a que, en ocasiones, el conocimiento que se está representando se puede implicar en diferentes áreas al pertenecer más a un conocimiento relacionado con procesos.

Hace posible analizar el conocimiento de un campo, por ejemplo en lo que se refiere al estudio de los términos y relaciones que lo configuran ya sea formalmente o no.

2.3.4 ONTOLOGÍAS: Descripción General.

En este apartado se explica muy brevemente una tecnología imprescindible para conseguir la Web Semántica: las ontologías.

Con las ontologías, los usuarios organizan la información de manera que los agentes de software podrán interpretar el significado y, por tanto, podrán buscar e integrar datos mucho mejor que ahora. Con el conocimiento almacenado en las ontologías, las aplicaciones podrán extraer automáticamente datos de las páginas web, procesarlos y sacar conclusiones de ellos, así como tomar decisiones y negociar con otros agentes o personas. Por ejemplo: mediante las ontologías, un agente encargado de comprar viviendas se podrá comunicar con agentes hipotecarios (de entidades bancarias) y con agentes inmobiliarios (de empresas constructoras e inmobiliarias).

Las ontologías proceden del campo de la Inteligencia Artificial; especifican vocabularios comunes para las personas y aplicaciones que trabajan en un dominio. Toda ontología representa cierta visión del mundo con respecto a un dominio.

Cualquier persona tiene en su cabeza ontologías mediante las que representa y entiende el mundo que lo rodea. Estas ontologías no son explícitas, en el sentido de que no se detallan en un documento ni se organizan de forma jerárquica o matemática.

Las máquinas carecen de las ontologías con las que nosotros contamos para entender el mundo y comunicarse entre ellas; por eso necesitan ontologías explícitas.

Las ontologías explícitas se pueden expresar de muchas maneras. Como mínimo, deben incluir un vocabulario de términos, con la definición de cada uno. Las ontologías sencillas suelen representarse como una jerarquía de conceptos relacionados y ordenados.

2.4 Trabajos actuales sobre generación de ontologías

Existen algunos trabajos sobre la generación de diccionarios semánticos. Básicamente existen 3 tipos de generación de ontologías.

2.4.1 Generación manual de ontologías

Este tipo de generación de ontologías resulta ser muy caro en cuanto a recursos, ya que para generar una ontología manualmente es necesario contar con uno o varios expertos en el dominio donde se pretende generar la ontología, estos expertos también deben conocer los lenguajes de definición de ontologías y saber utilizar las herramientas para la construcción de diccionarios semánticos (Guarino, 1997).

2.4.1.1 Herramientas

En los últimos años han aparecido muchos entornos para la construcción de ontologías. Estas se organizan en las siguientes categorías (Asuncion, 2002):

- Desarrollo de ontologías.
- Evaluación de ontologías.
- Combinación e integración de ontologías.
- Herramientas de anotación basadas en ontologías.
- Almacenamiento y consulta de ontologías
- Aprendizaje sobre ontologías.

Dentro del primer grupo de herramientas (Desarrollo de ontologías) las mas importantes son las siguientes:

2.4.1.1.1 Ontolingua

Esta herramienta fue creada por el *Knowledge Software Laboratory* (KSL) de la Universidad de Stanford, OntoLingua provee de un medio ambiente para buscar, crear, editar, modificar, y usar ontologías. Soporta más de 150 usuarios activos (Sitio web de Stanford). También incluye una API para integrar ontologías del servidor con agentes preparados para internet.

2.4.1.1.2 Protégé

Protégé es una plataforma de código abierto y libre que proporciona herramientas para construir diccionarios semánticos a partir de modelos de un dominio, para aplicaciones basadas en conocimiento (Sitio web de Protege). Protégé implementa un gran conjunto de estructuras para el modelamiento del conocimiento que soportan la creación, visualización, y manipulación de diccionarios semánticos

representándolos en varios formatos. Además, Protégé puede ser extendido por medio de una arquitectura de plug-ins y una interfaz de programación de aplicaciones basada en Java, para la construcción de herramientas y aplicaciones. Protégé soporta dos maneras de modelar diccionarios semánticos: el editor Protégé-Frames y el editor Protégé-OWL. El primero permite construir diccionarios semánticos basados en frames o estructuras de modelamiento, de acuerdo con el protocolo abierto de conectividad basado en conocimiento OKBC (Open Knowledge Base Connectivity). El segundo permite construir los diccionarios semánticos para la Web Semántica, en particular en el lenguaje de Ontologías Web OWL (Ontology Web Language) de la W3C (World Wide Web Consortium).

El Editor Protégé-OWL permite:

- Cargar y guardar diccionarios semánticos OWL.
- Editar y visualizar clases, propiedades y reglas.
- Definir características de clases lógicas como expresiones OWL.
- Ejecutar razonadores como clasificadores lógicos de descripción.

2.4.1.2 Lenguajes

2.4.1.2.1 OWL

Este lenguaje de definición de diccionarios semánticos fue creado por el World Wide Web Consortium (W3C), OWL (Lenguaje de ontologías Web) está diseñado para usarse en aplicaciones que necesitan procesar el contenido de la información en lugar de la simple presentación de información para los humanos (sitio web de W3C). OWL facilita la interpretación de contenido web a maquinas que soportan XML, RDF y RDF-Shema (RDF-S) dándole un vocabulario adicional con una semántica formal. OWL cuenta con 3 sub-lenguajes: OWL Lite, OWL DL y OWL Full.

OWL es un lenguaje para representar el conocimiento de manera descriptiva, basado en lógica. OWL nos permite definir un diccionario semántico expresándose en lenguaje XML (Sitio web de OWL).

Un diccionario semántico OWL puede incluir clases, propiedades y sus instancias. La semántica formal de OWL especifica como derivar sus consecuencias lógicas. Por ejemplo, hechos no presentados literalmente en el diccionario semántico, pero detallada en la semántica, estos detalles pueden estar en un solo documento, o en múltiples documentos que fueron combinados usando mecanismos definidos con OWL.

2.4.1.3 Metodologías de construcción

Existen también muchas metodologías de desarrollo que los expertos en ontologías deben conocer para desarrollar eficientemente una ontología. Mencionaremos algunas.

2.4.1.3.1 TOVE (Toronto Virtual Enterprise)

Esta metodología está inspirada en el desarrollo de sistemas basados en conocimiento usando el Primer Orden Lógico (Grüninger & Fox, 1995). Se propone que se sigan lo siguientes pasos:

1. Identificar intuitivamente los principales escenarios de uso, es decir, posibles aplicaciones donde se usará la ontología.

2. Identificar un conjunto de preguntas en lenguaje natural (llamadas preguntas de aptitud), que serán usadas para determinar el alcance de la ontología.

3. Estas preguntas y sus respuestas son usadas para extraer los conceptos principales y sus propiedades, relaciones y axiomas en la ontología.

4. Definir las definiciones y limitaciones en la terminología, de ser posible. Las especificaciones son representadas en Primer Orden Lógico e implementadas en Prolog.

5. Probar la competencia de la ontología generando teoremas de integridad con respecto a las preguntas de aptitud.

2.4.1.3.2 Methontology

Methontology es una metodología creada en el Laboratorio de Inteligencia Artificial de la Universidad Técnica de Madrid. La creación de la ontología puede empezar desde cero o en base a la reutilización de otras existentes. Methontology incluye la identificación del proceso de desarrollo de la ontología (calendario, control, aseguramiento de calidad, adquisición de conocimiento), un ciclo de vida basado en la evolución de prototipos, para lo cual se sigue -los pasos definidos en el estándar IEEE 1074 de desarrollo de software (Corcho, Fernández-López, & Gómez-Pérez, 2003): que son:

- Especificación.- Definir el alcance y granularidad de la ontología.

- Conceptualización.- Permite organizar y estructurar el conocimiento adquirido mediante tablas, lenguaje UML, jerarquías etc.

- Implementación.- Representa la formalización de la ontología; es decir pasar la conceptualización de la ontología a un lenguaje como RDF, OWL, etc.

- Evaluación.- Comprobar el funcionamiento de la ontología.

2.4.1.3.3 Otras metodologías

Las reglas generales para el diseño de ontologías pueden ser las siguientes (Noy & McGuinness, 2000):

1. No existe una manera “correcta” de modelar un dominio, siempre hay varias alternativas, la mejor solución siempre depende en cómo se va a usar la ontología.

2. El desarrollo de ontologías es un necesariamente un proceso iterativo.

3. Los conceptos en la ontología deben ser objetos (físicos o lógicos) cercanos a las relaciones en el dominio de interés. Los conceptos pueden ser sustantivos y las relaciones verbos, en oraciones que describan el dominio.

Y los pasos generales para generar las ontologías pueden ser los siguientes (Noy & McGuinness, 2000):

1. Determinar el dominio y alcance de la ontología.

2. Considerar reutilizar ontologías.

3. Enumerar los términos importantes en la ontología.

4. Definir las clases y las jerarquías de clases.

5. Definir las propiedades de las clases.

6. Definir la cardinalidad de las propiedades.

7. Crear instancias.

2.4.2 Generación semiautomática de ontologías.

Existen varios trabajos sobre generación semiautomática de diccionarios semánticos, en este apartado mencionaremos algunos.

2.4.2.1 Text-to-Onto

Soporta la creación de semiautomática de diccionarios semánticos usando algoritmos de minería de textos (Sitio de TexttoOnto). Extrae las estructuras

conceptuales usando la frecuencia de términos del texto, la jerarquía de conceptos es extraída usando unos algoritmos de agrupación jerárquica y las relaciones no taxonómicas son extraídas usando algoritmos de minería de asociación de reglas. Este sistema requiere la verificación del usuario en cada etapa del proceso de extracción de ontologías. Este trabajo funciona con textos escritos en alemán.

2.4.2.2 Hasti

Opera de modo cooperativo y no supervisado. En el modo cooperativo el usuario decide la selección o rechazo en cada etapa del proceso. Por ejemplo: el usuario tiene que seleccionar conceptos de los candidatos. En el modo no supervisado el sistema automáticamente selecciona cada componente de la ontología. Inicialmente acepta unos cuantos conceptos y relaciones taxonómicas como elementos centrales y posteriormente extiende estas semillas agregando más conceptos. Este trabajo funciona con textos escritos en persa.

2.4.3 Generación automática de ontologías

2.4.3.1 Método de Punuru.

Punuru J., desarrolla un conjunto de técnicas para la extracción automática de componentes ontológicos de textos de un dominio. Estas técnicas combinan bases de conocimientos léxicos como WordNet, estadísticas y heurísticas, procesamiento de texto entre otros métodos de adquisición de conocimiento. Estos métodos son independientes del dominio en el que se esté trabajando y extrae cada componente automáticamente.

En su trabajo una frase es considerada como relevante si alguna o todas las palabras están en la lista inicial de términos.

Dependiendo de la cantidad de sentidos (según WordNet) de un término, entre menor sea esa cantidad el término pertenece más al dominio. Si ya se encontró una frase que pertenece al dominio, todas las que tengan la misma palabra de terminación también se considera del dominio. Si ya se encontró una frase que

pertenece al dominio todas las frases que tengan una o más palabras de esta frase también se consideran del dominio.

Una palabra tiene varios sentidos o significados dependiendo el contexto.

2.4.3.2 OntoLearn

Es un método de extensión de ontologías basado en minería de textos y técnicas de aprendizaje automatico. OntoLearn empieza con una ontología genérica existente como WordNet u otras, y un conjunto de documentos de un dominio dado. Produce una extensión de la ontología original pero aplicada al dominio dado (Gómez-Pérez, Fernandez-López, & Corcho, 2004).

3 MÉTODO PROPUESTO

3.1 Arquitectura del método propuesto

El método que proponemos en esta tesis consta de 3 fases, tal como se muestra en el siguiente diagrama.

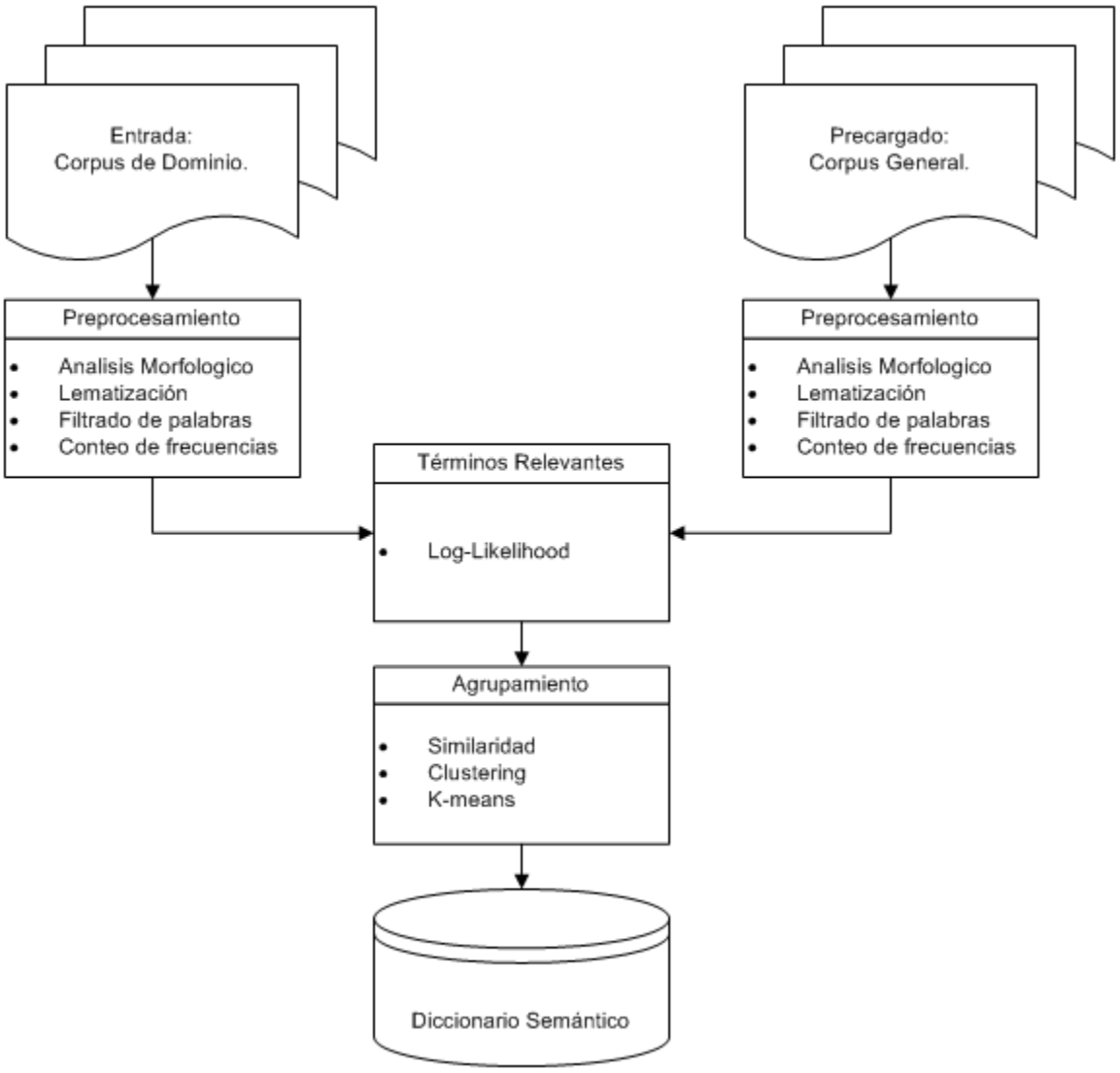


Figura 1 Diagrama del método propuesto

La arquitectura del método propuesto es la siguiente:

Se toma como entrada al sistema el corpus de un dominio. Siendo un corpus una serie de documentos referentes a un dominio en particular.

A este corpus se le hace un pre-procesamiento, en el cual hacemos un análisis morfológico, es decir, se obtiene la información gramatical de las palabras, luego se eliminan las palabras denominadas stopwords, es decir, palabras que no incluyen información al ningún dominio, y a las palabras restantes les hacemos una lematización, por ejemplo, las palabras, trabaja, trabajamos, trabajó, en su información gramatical tienen como lema a la palabra trabajar, así que al momento de hacer los conteos de frecuencias, estas palabras solo a cuentan a la palabra trabajar.

Con base con las frecuencias de las palabras lematizadas, hacemos unos cálculos y comparaciones con un corpus general previamente pre-procesado, para encontrar los términos relevantes en el dominio dado.

Estos términos, los agrupamos en clústeres, donde los términos pertenecientes a un clúster tienen alta similaridad y los términos tienen baja similaridad con términos de otros clústeres.

3.2 Preprocesamiento

La fase de pre-procesamiento es el proceso de extraer los términos que contienen significado para el dominio dado, y se hace un conteo de frecuencias mientras leemos los archivos de entrada. En este proceso se reformatea y se filtra la información para hacer un mejor análisis estadístico.

Para los fines de esta tesis tomaremos en cuenta las siguientes restricciones para el análisis de textos:

- No se toman en cuenta los números, y marcas de puntuación.

- Los signos de puntuación al final de una palabra se deben eliminar, por ejemplo: cuando en el texto se encuentre “procesador,” solo tomará en cuenta “procesador”.

- No se toman en cuenta las letras en mayúsculas, es decir, todo el texto será cambiado a minúsculas. Por ejemplo: “Procesador” será transformado en “procesador”.

- Se eliminan las palabras que no aportan información para un dominio, tales como artículos, preposiciones, etc.

- Se hace el conteo de frecuencias en base a las palabras con el mismo lema (lematización) de modo que palabras como *trabaja*, *trabajar*, *trabajamos*, suman a la frecuencia de la palabra *trabajar*.

Al finalizar esta fase tenemos una tabla conteniendo, para cada palabra lematizada, su frecuencia por documento (*Term Frecuency*).

Palabra	Documento	Frecuencia
software	14	3
software	16	3
software	20	12
software	21	6
software	24	7
software	25	1
software	28	2
software	30	3
software	31	1
software	33	1
software	35	1
software	38	1
software	39	2

software	41	5
software	42	2
software	43	1
software	45	2
software	47	64
software	48	4
software	49	3
software	50	1
software	52	2
software	54	2

Tabla 3-1 Fragmento de la tabla de Frecuencias

En la tabla 3-1 se muestra un fragmento de la tabla de frecuencias del Corpus de dominio en informática, donde se puede apreciar que la palabra software no está contenida en todos los documentos y que en algunos documentos aparece varias veces.

3.3 Extracción de conceptos

El trabajo de preprocesamiento se aplica a 2 corpus, uno general y otro de dominio, para tener información estadística de frecuencias de palabras cuando el corpus es un corpus de manera general (tal como se habla normalmente), y frecuencias de palabras cuando el corpus es referente a un dominio (tal como hablan los expertos en un domino especifico).

La idea de los 2 corpus es debido a que en un documento informal o en una plática normal, se utiliza un vocabulario distinto que cuando se está escribiendo un documento de un tema específico, o se está impartiendo una conferencia.

Con esta información estadística se obtienen frecuencias de palabras y se aplica el algoritmo de likelihood explicado en el capítulo 2.

Palabra	Frec. Dominio	Frec. General	Frec. E. Dominio	Frec. E. General	Log-Likelihood
socket	1	0	0.010286744	0.989713252	9.153798103
sofisticado	5	169	1.789893508	172.2101135	3.912798405
soft	1	12	0.13372767	12.86627197	2.351035118
software	430	831	12.97158432	1248.028442	2334.961182
softwares	2	1	0.030860232	2.969139814	14.50919151
sofware	2	2	0.041146975	3.958853006	12.8037796
sol	2	933	9.618105888	925.381897	-9.016687393
solamente	20	1714	17.83721352	1716.162842	0.254846573

Tabla 3-2 Fragmento de la tabla de cálculo de Log-Likelihood

En la tabla 3-2 se muestra un conjunto de palabras, con los cálculos correspondientes para el dominio de informática. Donde:

- El campo Frec. Dominio y Frec. General son las frecuencias observadas (Term Frecuency) de una palabra en el corpus de dominio y en el Corpus General respectivamente.

- El campo Frec. E. Dominio y Frec. E. General son las frecuencias esperadas de una palabra en el corpus de dominio y en el Corpus General respectivamente. Estos valores fueron calculados con las siguientes fórmulas:

$$E_d = c * \frac{a+b}{c+d} \quad \text{Y} \quad E_g = d * \frac{a+b}{c+d}$$

Fórmula (15) Cálculo de Frecuencia esperada

Donde:

E_d y E_g son los valores esperados en el corpus de dominio y en el corpus General respectivamente.

c y d es el total de palabras en el corpus de dominio y en el corpus General respectivamente.

a y b es la Frecuencia (Term Frecuency) de la palabra en el corpus de Dominio y en el corpus General respectivamente.

- El campo Log-Likelihood es la ponderación obtenida al aplicar la formula de Log-Likelihood.

$$G_2 = 2 * \left(\left(a * \log\left(\frac{a}{E_d}\right)\right) + \left(b * \log\left(\frac{b}{E_g}\right)\right)\right)$$

Fórmula (16) Calculo de Log-Likelihood

Donde:

E_d y E_g son los valores esperados en el corpus de Dominio y en el corpus General respectivamente.

a y b es la Frecuencia (Term Frecuency) de la palabra en el corpus de Dominio y en el corpus General respectivamente.

Nótese que en la tabla 3-2 la palabra “software” tiene un Log-Likelihood de 2334.3961182, la palabra “sol” tiene un log-Likelihood de -9.01668793, es decir las palabras muy relacionadas al dominio, resultan con una ponderación alta. La palabra “sofware" que está mal escrita también tiene su propia ponderación, esto refleja el hecho de que este método es sensible a faltas de ortografía y a la mala elección de los elementos del corpus.

3.4 Medidas de similaridad

La medida de similaridad utilizada en esta tesis es la similaridad por coseno del ángulo. Esta medida de similaridad la utilizaremos al momento de hacer el clustering. En la documentación revisada en el estado del arte, la medida de

similaridad es una matriz cuadrada, donde están todas las palabras y en la diagonal todos los valores son 1, porque es donde se intersectan una palabra con ella misma.

Para esta tesis simulamos esta matriz con una tabla con 3 campos, donde los primeros 2 campos son de palabras y el ultimo campo es el valor de la similaridad entre estas 2 palabras, calculando en base a las frecuencias en el corpus de dominio. Utilizando la siguiente fórmula:

$$\cos(x, y) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Fórmula (17) Distancia por coseno

Donde:

n es la cantidad de documentos en el dominio.

x_i es la frecuencia de la palabra x en el documento i.

y_i es la frecuencia de la palabra y en el documento i.

Palabra1	Palabra2	Similaridad
Software	algoritmo	0.032
Software	algoritmos	0.057
Software	almacén	0.018
Software	almacenamiento	0.044
Software	almacenar	0.086
Software	aplicación	0.420
Software	archivo	0.203
Software	arpanet	0.031
Software	arquitectura	0.271
Software	artificial	0.029

Software	basar	0.119
Software	bioinformática	0.073
Software	cálculo	0.055
Software	característica	0.220
Software	ciencia	0.071
Software	circuito	0.018
Software	código	0.759
Software	compilador	0.185
Software	componente	0.171

Tabla 3-3 Fragmento de la tabla de Similaridad entre palabras (por orden alfabético)

En la tabla (3-4) se aprecia la similaridad por coseno del ángulo entre la palabra software y otras del dominio de informática, se observa que entre software y código es de 0.759238513 y entre software y bioinformática es de sólo 0.073920062.

3.5 Agrupamiento.

El método utilizado en esta tesis para agrupar los términos, es el llamado algoritmo k-means.

Este algoritmo recibe como parámetro un numero k, que es la cantidad de grupos que se obtendrán al final del algoritmo.

El procedimiento es el siguiente:

- Elegimos k términos como centro de los k clúster.
- Cada término restante es asignado al clúster cuyo centro tenga un mayor índice de similaridad con éste término (Véase apartado anterior).
- Se hace un reajuste para encontrar el nuevo centro del clúster.
- Se repiten los 2 pasos anteriores hasta que el centro no cambie.

Al final se obtienen clústeres con la siguiente estructura:

Centro	Elementos	Similaridad con el Centro
describir	describir solucionar(0.72958144294236), arquitectura(0.530256444275185), matemática(0.509922398646436), interfaces(0.449127543840738), patrón(0.439137655344973), diseñar(0.427533841333521), patrona(0.367712282463181), diseñador(0.335131368552743),	1
disco	almacenamiento(0.872512216486504), cd(0.928569200464337), discar(0.861329741127726), disco(1), disquetes(0.93449370145437), dvd(0.947514248563855), flash(0.877319009647372), rom(0.933910891781232), soportar(0.591941313199565), usb(0.89953182776237), velocidad(0.753637402422804), óptico(0.840082558882581),	
ejecutar	archivo(0.610302943145724), buffer(0.671762878661979), e/s(0.661403665235961), ejecutar(1), instrucción(0.795692688275484), interrupción(0.629444547042457), llamada(0.735845563812425), operativo(0.77822950335828), paginación(0.268898828370022), procesador(0.6775406236865), procesadores(0.473641002419669), proceso(0.598212535376625), programa(0.77927069053853), traducción(0.319736228476198), virtual(0.345041365922363),	
electrónica	analizador(0.719268335724235), analógicos(0.919183310979738),	

Centro	Elementos	Similaridad con el Centro
	combinacionales(0.307793505625546), electrónica(1), electrónico(0.655469214258043), eléctrico(0.820817512788439), miniaturización(0.760882701922802), voltaje(0.756307048667989), válvula(0.812132190762674), –(0.668074315642542),	
eniac	análogo(0.959681464731238), aritmética(0.792904928003396), babbage(0.97947079341596), calculador(0.99628594668259), cinta(0.828455118621231), colossus(0.998868137724437), computer(0.824024172645483), cálculo(0.91434136758329), digitales(0.624172273100834), eniac(1), ibm(0.951302988308988), magnético(0.764062306877658), mecánica(0.895906919000422), máquina(0.843032554753135), neumann(0.870986364878406), operacional(0.713477241231741), perforar(0.995860011795163), relés(0.966969210705448), tubo(0.97776253922769), univac(0.856522361818368), von(0.913451487686955), z3(0.779984973495458), zuse(0.976695676411961), ábaco(0.993910945151609),	

Tabla 3-4 Fragmento de algunos clústeres del dominio de informática.

4 EXPERIMENTOS Y RESULTADOS

4.1 Experimento 1:

4.1.1 Descripción

Para este experimento se utilizaron los siguientes corpus:

Corpus de comparación: son las noticias del periódico Excélsior, 247 archivos, con 11, 936,114 palabras en total.

Corpus de dominio: utilizamos algunas páginas de wikipedia relacionadas al área de informática, tales como, informática, software, programación, etc. Son 72 archivos con 142,132 palabras.

Por lo tanto, el diccionario semántico resultante será para el dominio de Informática.

4.1.2 Resultados

En la segunda fase, la de extracción de términos, los 15 términos más relevantes del dominio de informática, y su respectiva ponderación con el algoritmo log-likelihood, fueron los siguientes.

Palabra	Log-likelihood
Dato	2543
Sistema	2345
Software	2279
Computador	1713
Lenguaje	1438
Maquina	1371

Palabra	Log-likelihood
Memoria	1314
Información	1039
Red	1006
Programación	984
Dispositivo	900
Ordenador	854
Diseñar	741
Usar	680
utilizar	668

Tabla 4-1 Los 15 Términos más relevantes en el área de informática

En la tercera fase, la extracción de relaciones semánticas se obtuvo como ejemplo los siguientes clústeres:

Centro	Elementos
Algoritmo	Algoritmo, algoritmos, array, for, implementación, implementar.
Analógica	Analógica, binario, voltaje.
B2B	B2B, business, cliente, consistir, electrónico, hosting, internet, servidor.
Código	Compilador, compiladores, compuesto, código, lenguaje, maquina, programa.
Formato	Archivar, archivo, audio, avi, codificar, compresión, especificación, estándar, formato, formatos, informático, mov, video.

Tabla 4-2 Ejemplo de los clústeres obtenidos

4.2 Experimento 2:

4.2.1 Descripción

Para este experimento se utilizaron los siguientes corpus:

Corpus de comparación: son las noticias del periódico Excélsior, 39 archivos, con 1,365, 991 palabras en total.

Corpus de dominio: utilizamos algunas páginas de wikipedia relacionadas al área de informática, tales como, informática, software, programación, etc. Son 26 archivos con 44, 495 palabras.

Por lo tanto, el diccionario semántico resultante será para el dominio de Informática.

Se toma el dominio de informática porque el autor de esta tesis es el tema que más domina, para que al momento de evaluar el método, esto sea más sencillo. Pero se puede tomar cualquier otro conjunto de textos de otros dominios.

4.2.2 Configuración

La configuración para este experimento fue la siguiente:

Entrada: 26 archivos del área de informática tomados de wikipedia, que formaron el corpus de dominio. 39 archivos tomados de las noticias del periódico Excélsior, que formaron el corpus general

Conceptos: Se tomaron sólo los 400 conceptos mejor ponderados por el algoritmo Log-likelihood

Clústeres: el algoritmo k-means fue configurado para proporcionar 19 clústeres con los conceptos seleccionados.

4.2.3 Ponderación por log-likelihood

En la segunda fase, la de extracción de términos, obtuvimos 400 términos relacionados con el área de informática al ordenarlos por la ponderación obtenida con el método log-likelihood de mayor a menor, listarlos todos abarca mucho

espacio, por esta razón la tabla 4-3 sólo muestra los 15 términos más relevantes del dominio de informática, y su respectiva ponderación con el algoritmo log-likelihood.

Palabra	Log-likelihood
Dato	1506
Computador	863
Circuito	467
Memoria	384
Señal	372
Secuencia	353
Computación	351
Información	346
Dispositivo	342
Algoritmos	341
Electrónico	322
Basar	319
Diseñar	307
Algoritmo	288
Utilizar	282

Tabla 4-3 Los 15 Términos más relevantes en el dominio de informática

Al observa la tabla 4-3 y ver los resultados obtenidos, podemos observar que los términos seleccionados están relacionados al área de informática.

4.2.4 Clusterización

En la tercera fase, la extracción de relaciones semánticas, aplicamos el algoritmo con los siguientes parámetros, una k de 19 para el algoritmo k-means, es decir 19 clústeres, y utilizando los 400 términos extraídos con el algoritmo log-likelihood, con estos parámetros se obtuvo los siguientes clústeres:

Centro	Elementos	Similaridad con el Centro
algoritmo	algoritmo	1
	for	0.963
	algoritmos	0.860
	implementación	0.781
	array	0.774
	implementar	0.681
	árbol	0.255
analógica	analógica	1
	voltaje	0.849
	binario	0.472
as	as	1
	if	1
	int	1
	integer	1
	pseudocódigo	1
	return	1
	vtemp	1
	diagrama	0.981
	descripción	0.946
	turing	0.894
	end	0.801
b2b	b2b	1
	business	1
	hosting	1
	cliente	0.928
	servidor	0.928
	internet	0.872
	to	0.872
	electrónico	0.849
	consistir	0.686

Centro	Elementos	Similaridad con el Centro
biología	biología	1
	bioinformática	0.999
	adn	0.998
	alineamiento	0.998
	bioinformáticos	0.998
	clustalw	0.998
	fago	0.998
	gen	0.998
	genoma	0.998
	genomas	0.998
	genome	0.998
	genómica	0.998
	génica	0.998
	homología	0.998
	human	0.998
	microarrays	0.998
	modelado	0.988
	nucleótidos	0.988
	predicción	0.998
	proteína	0.998
	proteína-proteína	0.998
	sanger	0.998
	secuenciación	0.998
	evolutivo	0.991
	secuencia	0.991
	biológico	0.987
	computacional	0.925
	protocolo	0.901
	variedad	0.882
	análisis	0.880
	técnica	0.873
	estructura	0.871
	interacción	0.871
	completar	0.815
	montaje	0.800
	herramienta	0.788
	menudo	0.787
	usar	0.783
	talar	0.762
	software	0.749
	visualizar	0.749
	cuantificación	0.706
	modelo	0.702
	automatizar	0.691
	búsqueda	0.626

Centro	Elementos	Similaridad con el Centro
componente	componente	1
	transistor	0.878
	tubo	0.860
	funcionar	0.816
	conexión	0.845
	dispositivo	0.843
	etc	0.807
	tecnología	0.793
	digitales	0.759
	microprocesadores	0.751
	velocidad	0.743
	lógica	0.668
	soler	0.578
	altavoz	0.564
computación	computación	1
	ciencia	0.980
	constable	0.976
	científica	0.971
	cómputo	0.970
	disciplina	0.939
	matemática	0.877
	usualmente	0.869
	teoría	0.851
	computacionales	0.806
	ingeniería	0.780
	estudiar	0.770
	artificial	0.739
	matemático	0.717
	informática	0.672
	paralelo	0.595
	programación	0.579

Centro	Elementos	Similaridad con el Centro
conjunto	conjunto	1
	notación	0.945
	problema	0.906
	finito	0.881
	binaria	0.862
	complejidad	0.845
	np	0.824
	np-completo	0.824
	número	0.822
	tamaño	0.787
	elemento	0.759
	coste	0.735
	lineal	0.684
	comúnmente	0.616
	montículo	0.103
código	código	1
	compilador	0.969
	compiladores	0.956
	lenguaje	0.841
	máquina	0.845
	programa	0.614 0.404
	compuesto	
descifrar	descifrar	1
	criptografía	0.999
	cifrar	0.999
	método	0.760
	texto	0.700
	denominar	0.645

Centro	Elementos	Similaridad con el Centro
dimensión	dimensión	1
	cubo	0.996
	espacial	0.977
	almacén	0.995
	marts	0.995
	metadato	0.995
	middleware	0.995
	warehouse	0.995
	data	0.994
	olap	0.968
	tabla	0.891
	operacional	0.890
	variable	0.875
	definición	0.846
	especificar	0.773
	usuario	0.765
	poseer	0.758
	almacenar	0.747
	dato	0.739
	colección	0.687
	arquitectura	0.519
	registro	0.201
diseñar	diseñar	1
	diseñador	0.988
	objetar	0.816
	funcional	0.765
	procesar	0.723
	proceso	0.696
formato	formato	1
	avi	0.993
	compresión	0.993
	especificación	0.993
	formatos	0.993
	mov	0.993
	archivar	0.992
	vídeo	0.985
	audio	0.968
	archivo	0.946
	informático	0.802
	codificar	0.686
	estándar	0.551

Centro	Elementos	Similaridad con el Centro
periférico	periférico	1
	mouse	0.977
	memoria	0.975
	cpu	0.962
	microprocesador	0.953
	disco	0.949
	ram	0.943
	procesador	0.943
	usb	0.934
	discar	0.932
	portátil	0.918
	tarjeta	0.914
	hardware	0.913
	monitor	0.907
	almacenamiento	0.906
	placa	0.904
	isbn	0.896
	dram	0.896
	pines	0.896
	celda	0.860
	procesamiento	0.859
	«	0.859
	»	0.859
	teclado	0.844
	inglés	0.845
	usado	0.843
	óptico	0.832
	gráfico	0.824
	computadores	0.77
	introducción	0.751
	pc	0.728
	módulo	0.691
	computador	0.683
	ratón	0.674
	pantalla	0.673
	impresora	0.651
	típico	0.619
	escáner	0.608
	procesadores	0.603
	instrucción	0.494
	entrada/salida	0.455
	e/s	0.439
	operandos	0.222
	acumulador	0.179

Centro	Elementos	Similaridad con el Centro
potencia	potencia1	1
	válvula	0.979
	analógicos	0.952
	semiconductor	0.927
	corriente	0.921
	alternar	0.917
	analizador	0.898
	electrónica	0.861
	conmutación	0.798
	eléctrico	0.790
	sonido	0.439
	pila	0.223
	supercomputadoras	0.168

Centro	Elementos	Similaridad con el Centro
red	red1	1
	principal	0.876
	artículo	0.872
	permitir	0.864
	utilizar	0.855
	vario	0.849
	aplicación	0.833
	información	0.801
	través	0.797
	tipo	0.785
	sistema	0.781
	ejemplo	0.759
	característica	0.746
	interfaz	0.730
	forma	0.725
	gestión	0.724
	operativo	0.716
	acceder	0.714
	diferente	0.713
	basar	0.682
	contener	0.673
	operación	0.663
	función	0.656
	clasificar	0.618
	ordenador	0.614
	ejecutar	0.569
	programador	0.56
	cálculo	0.552
	modelar	0.548
	relacionales	0.516
	interfaces	0.461
	objeto	0.434
	relacional	0.406
rápido	rápido	1
	acceso	0.902
	sencillo	0.802
	soportar	0.787
	web	0.734
	específico	0.559
	central	0.351
	fiabilidad	0.295
	paralelismo	0.209

Centro	Elementos	Similaridad con el Centro
señal	señal	1
	transductores	0.989
	transductor	0.980
	impedancia	0.949
	filtrar	0.940
	conversión	0.968
	acondicionamiento	0.90
	convertidor	0.908
	daq	0.908
	adquisición	0.907
	analógico	0.891
	conectar	0.885
	adaptación	0.882
	frecuencia	0.881
	medir	0.845
	tensión	0.825
	sensores	0.823
	digital	0.805
	cable	0.783
	control	0.761
	física	0.760
	entrada	0.757
	medición	0.739
	físico	0.727
	salida	0.721
	normalmente	0.655
	bus	0.617
	datar	0.385
térmico	térmico	1
	ci	0.980
	cápsula	0.980
	integration	0.980
	scale	0.980
	chip	0.942
	circuito	0.921
	chips	0.921
	integrar	0.919
	híbrido	0.895
	silicio	0.891
	reproductor	0.815
	amplificador	0.784
	fabricación	0.778

Tabla 4-4 Clústeres de la prueba 2

5 CONCLUSIONES

5.1 Conclusiones

Se presenta un sistema para la construcción automática de diccionarios semánticos usando la similitud distribucional observada en los textos de un dominio.

Este sistema construye diccionarios semánticos de un dominio en específico con las siguientes características:

- Contiene los términos más utilizados dentro de ese dominio.
- Pondera la semejanza entre los términos encontrados.
- Estos términos están agrupados en grupos con al menos un elemento, cuyos elementos tienen un alto grado de similitud.
- Estos términos tienen baja similaridad con los términos de otros clústeres.

BIBLIOGRAFÍA

(s.f.). Recuperado en Mayo de 2008, de wikipedia: http://es.wikipedia.org/wiki/Ontolog%C3%ADa_(inform%C3%A1tica)

(s.f.). Recuperado en Septiembre de 2008, de Sitio de TexttoOnto: http://sourceforge.net/projects/texttoonto

(s.f.). Recuperado en Mayo de 2008, de Sitio web de Protege: http://protege.stanford.edu/doc/users.html

(s.f.). Recuperado en Mayo de 2008, de Sitio web de Stanford: http://www.ksl.stanford.edu/software/ontolingua/

(s.f.). Recuperado en Mayo de 2008, de sitio web de W3C: http://www.w3.org/2001/sw/WebOnt/

(s.f.). Recuperado en Mayo de 2008, de Sitio web sobre ontologias: http://es.geocities.com/recupdeinformacion_ontologias/sobreontolgias.htm

(s.f.). Recuperado en Mayo de 2008, de Sitio web de OWL: http://www.w3.org/TR/owl-guide/

Alonso Fernández, A. M. (s.f.). Recuperado en Diciembre de 2009, de Sitio web de Andrés M. Alonso Fernández: http://www.uam.es/personal_pdi/ciencias/aalonso/Docencia/2003/ad-sl7.pdf

Asuncion, G.-P. (2002). A survey on ontology tools. *Universidad Politécnia de Madrid* .

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* .

Cimiano, P. (2006). *Ontology learning and population from text, algorithms, evaluation and applications.* New York, USA: Springer.

Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and Lenguajes for buildong ontologies. Wherte is theis meeting point? *Data & Knowledge Engineering 46* , 41-64.

Danger R., Sanz I., Berlanga R., Ruíz-Shulcloper J. (s.f.). CRISOL: Una aproximación a la generación automática de instancias para una Web Semántica.

Gómez-Pérez, A., Fernandez-López, M., & Corcho, O. (2004). *Ontological Engineering.* London: Springer Verlag.

Gruber, T. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Technical Report KSL-93-04* .

Grüninger, M., & Fox, M. (1995). Methodology for the design and evaluation of Ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing* .

Guarino, N. (1997). Understanding, building, and using ontologies. *International Journal of Human Computer Studies* .

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th International Conference on Computational Linguistics (COLING)* .

Imsombut, A., & Kawtrakul, A. (2007). Automatic building of an ontology on the basis of text corpora in Thai.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *: Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1 (Univ. of Calif. Press, 1967),.*

Maedche, A., Volz R.,. (California, USA). The ontology extraction & maintenance framework text-to-onto. *IEEE International Conference on Data Mining* , 2001.

Maedche, A; S., Staab. (2000). Discovering conceptual relations from text. *Proceedings of ECAI* .

Noy, N. F., & McGuinness, D. L. (2000). Ontology Development 101: A Guide to Creating Your First Ontology.

Pantel, P., & Ravichandran, D. (2004). Automatically labeling semantic classes. En *Human language technologies and north american association of computational linguistics.*

Punuru, J. (2007). *Knowledge-based methods for automatic extraction of domain-specific ontologies.*

Rayson, P., & Garside, R. (2000). Comparing Corpora using Frequency Profiling. *Proceedings of the workshop on Comparing corpora - Volume 9* .

Sánchez-Cuadrado, S.; J., LLoréns; J., Morato; A., Hurtado J. (s.f.). Extracción automática de relaciones semánticas.

Uschold, M. (1996). Building Ontologies: Towards a unified Methodology. *Preceedings of Expert Systems '96, the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems, held in Cambridge, UK* .

Uschold, M., & Grunninger, M. (1996). Ontologies: Principles Methods and Applications. *Knowledge Egineering Review* .