UNIVERSIDAD AUTÓNOMA METROPOLITANA UNIDAD AZCAPOTZALCO

UNIDAD AZCAPOTZALCO

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

PROYECTO DE INVESTIGACIÓN

MEDICIÓN AUTOMÁTICA DE LA SIMILITUD ENTRE TEXTOS USANDO CARACTERÍSTICAS INDEPENDIENTES DEL IDIOMA

LICENCIATURA EN INGENIERÍA EN COMPUTACIÓN

2016- INVIERNO

PRESENTA

PAUL ERIK SORIANO LAGUNA

210329204

ASESOR: MARICELA CLAUDIA BRAVO CONTRERAS

CO-ASESOR: JOSÉ ALEJANDRO REYES ORTÍZ

MÉXICO, D.F. 04 ABRIL DE 2016

Yo, Maricela Claudia Bravo Contreras, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

MARICELA CLAUDIA BRAVO CONTRERAS

Yo, José Alejandro Reyes Ortíz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

JOSÉ ALEJANDRO REYES ORTÍZ

Yo, Paul Erik Soriano Laguna, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

PAUL ERIK SORIANO LAGUNA

Resumen

Con el incremento tan acelerado de la información disponible en medios digitales se crea la necesidad de emplear nuevas herramientas de software que permitan procesar la información a mayor velocidad. Sin importar cual sea la finalidad (detección de plagio, relación de temas, similitud de contenido, etc.), obtener automáticamente la similitud textual que existe en un par de documentos escritos en el mismo idioma, supone un ahorro de esfuerzo y tiempo.

Las herramientas que permiten conocer la similitud que existe entre textos se pueden emplear sobre publicaciones científicas o noticias contenidas en la Web con la finalidad de saber en qué medida son parecidos un par de textos sin tener que revisar el documento completo.

En el presente trabajo se hace uso de métricas, así como la fórmula de la distancia del coseno para determinar el grado de similitud que existe en un par de documentos de entrada, los cuales deben estar escritos ambos en el mismo idioma, ya sea inglés o español, pero no una combinación de ambos. Estos textos se procesan y se lematizan para obtener así la similitud que existe entre ambos basados en medición por n-gramas igual a uno. Finalmente, la entrada y los resultados obtenidos se presentan en una interfaz de programación web.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El manejo de grandes volúmenes de información se ha convertido en una tarea cotidiana para las empresas e instituciones hoy en día. Estas organizaciones generan, almacenan y recuperan información a diario la cual es utilizada para la toma de decisiones. La información que se encuentra en documentos es vasta mientras que la contenida en medios electrónicos va en aumento, la que circula en Internet tiene un crecimiento exponencial, cada día se generan nuevos contenidos en redes sociales, blogs, páginas Web, se divulgan noticias y se publican nuevas investigaciones.

Este vertiginoso aumento de disponibilidad de información provoca que su recuperación se vuelva una tarea complicada sin el uso de mecanismos adecuados que procesen los documentos. En el ámbito de las instituciones específicamente, el manejo de la información sobre las investigaciones es indispensable, además comúnmente se encuentra tanto en español como en inglés haciendo necesario procesar documentos en estos dos idiomas principalmente.

Dada esta problemática, se vuelve indispensable procesar automáticamente la información relacionada con el tema que está desarrollando cualquier investigador permitiendo analizar solo los documentos que tienen relación con su linea de investigación. Para este caso se emplea el Procesamiento de Lenguaje Natural [1](PLN). Desarrollando una herramienta que permite el análisis de información escrita proveniente de fuentes distintas, se pretende minimizar el impacto de la tarea de recuperación y análisis de la información.

La recuperación de la información se basa en el principio de obtener documentos relevantes ante la necesidad de un usuario de obtener información específica y no únicamente se refiere a encontrar patrones que sean correspondientes entre sí. La similitud semántica entre textos consiste en obtener una medida de semejanza entre estos. Esta medida debe estar basada en su significado o contenido semántico y se puede apoyar en listas de términos que figuren en dichos textos, por ello su comparación se vuelve un reto importante.

Este trabajo encuentra el grado de similitud entre dos textos cortos que pueden ser documentos de la Web o textos de una base de datos y escritos en diversos idiomas. Se usaron métricas para obtener la similitud que existe entre dos textos en el mismo idioma, posteriormente se muestra el resultado mediante una interfaz Web.

El trabajo se ha organizado de la siguiente forma: en la sección 3, se presentan los objetivos perseguidos en este proyecto terminal, en la sección 4, se expone la motivación

[1]Manning & Schütze. *Foundations of statistical natural language processing*, MIT Press. Cambridge. 1999

para la realización del mismo. En la sección 5 se presenta la teoría relacionada con similitud textual y recuperación de la información, en la sección 6 se presentan los módulos que integran el proyecto y la interfaz de presetación de resultados, en la sección 7 se presentan los resultados obtenidos, en la sección 8 presentamos las conclusiones importantes de este trabajo.

Capítulo 2

Motivos

2.1. Justificación

Los textos o documentos en la Web pueden estar escritos en diversos idiomas y con diversos formatos, esto convierte la tarea de similitud automática en una necesidad para las actividades cotidianas, tales como recuperación de información, clasificación, etc. También, los sistemas de detección de plagio, la recuperación de similitud y la clasificación, entre otros, requieren un módulo de similitud semántica para definir una distancia entre las palabras, o el uso de medios estadísticos para correlacionar palabras y contextos textuales de un corpus. La medición de similitud en textos usando características independientes del dominio, ayuda a las instituciones de investigación en las tareas de recuperación de información, comparación de textos y detección de plagio (español o inglés). Por lo tanto, en este trabajo se desarrollo una interfaz de programación de aplicaciones (API por sus siglas en inglés) para medir la similitud en textos que pueden estar en español o en inglés. Esta herramienta de software permite ingresar dos documentos y compararlos a través de dos métricas para así obtener su similitud, permitiendo recuperar los textos con contenido relacionado al artículo que se está escribiendo, beneficiando así en la reducción del tiempo de búsqueda que invierte el investigador.

2.2. Antecedentes

2.2.1. Trabajos Relacionados

Proyectos Terminales

1. Sistema de detección de plagio en archivos de texto. [1] Este proyecto realiza una comparación entre dos textos mediante el empleo de técnicas de procesamiento de lenguaje natural con la finalidad de determinar si estos son similares.

   El presente trabajo hace uso de una interfaz de programación de aplicaciones, el sistema de detección funciona sólo para dos documentos de texto y trabaja con texto ingresado por el usuario, con documentos de texto y con información obtenida de bases de datos.

2. Sistema de procesamiento de textos de investigación. [2] Este proyecto emplea técnicas de procesamiento de lenguaje aplicadas a textos, los textos pueden ser de diferente formato, como es .txt, .pdf, .doc, etc. Aplica un análisis a artículos científicos que se encuentran en idioma inglés.

   El presente trabajo permite analizar documentos que se encuentran en idioma inglés o en español, pero no deben estar combinados.

3. Sistema de agrupamiento de servicios Web semánticos utilizando un algoritmo bioinspirado. [3] Este trabajo realiza una comparación de similitud semántica en base a métricas. Aplica la prueba de similitud a las entradas y salidas de un servicio Web que obtiene a partir de ontologías. Además se basa en el algoritmo de agrupamiento inspirado en la colonia de hormigas.

   En el presente trabajo se realiza una comparación de la similitud semántica que se aplica a doumentos de texto de acuerdo a métricas.

Tesis

Identifying Similarity in Text: Multi-Lingual Analysis for summarization. [4] La tesis y el proyecto propuesto se relacionan en que ambos realizan un análisis de similitud en textos y que además estos pueden estar en dos idiomas. La diferencia es que los idiomas no son los mismos y que la tesis no aplica la comparación a documentos completos sino a enunciados cortos del texto que analiza mediante el uso y diseño de un sistema de procesamiento y no una API. El sistema que se propone en la tesis se basa en técnicas sencillas para la traducción de primitivas del lenguaje.

Artículos de investigación

Transactions on knowledge and data engineering. [5] El artículo se enfoca en realizar una búsqueda de similitud semántica en textos, la diferencia con el proyecto propuesto es que ésta va dirigida a textos que se encuentran en la Web y además lo hace para co-ocurrencias de fragmentos de párrafos o pares de palabras, mientras que el proyecto se realiza sobre textos completos, haciendo un cálculo de similitud mediante métricas y el artículo propone un método automático basado en un algoritmo.

Software

SEMILAR: A Semantic Similarity Toolkit.[6] Esta es una herramienta de software que permite determinar la similitud entre textos. Es posible obtener una licencia de uso de este software en su sitio Web. SEMILAR se encuentra disponible como una API con interfaz gráfica de usuario y al igual que el proyecto propuesto, se puede hacer uso de la herramienta en la Web ingresando texto. Otra similitud es que tanto la aplicación como el proyecto están basados en el lenguaje de programación java.

Capítulo 3

Objetivos

3.1. Objetivo general

Diseñar e implementar una interfaz de programación de aplicaciones con la finalidad de medir la similitud entre textos a partir de recursos heterogéneos utilizando características independientes del idioma..

3.1.1. Objetivos especificos

- Diseñar e implementar un módulo de extracción de textos a partir de recursos heterogéneos (Bases de datos, documentos, textos de la Web ingresados por usuarios) que servirán como base para la obtención de la similitud.
- Implementar tres métricas estadísticas existentes de similitud textual basadas en características independientes del idioma, es decir que funcionen con textos en español o inglés.
- Implementar aplicación web usando la interfaz de programación de aplicaciones, que desempeñe el proceso completo de obtención de la similitud textual con las tres métricas a partir de los textos (fuentes heterogéneas) de entrada con la finalidad de evaluar la efectividad del proceso.

Capítulo 4

Marco Teórico

4.1. Recuperación de la información.

El proceso de recuperación se lleva a cabo mediante consultas a la base de datos donde se almacena la información estructurada, mediante un lenguaje de interrogación adecuado. Es necesario tener en cuenta los elementos clave que permiten hacer la búsqueda, determinando un mayor grado de pertinencia y precisión, como son: los índices, palabras clave, tesauros y los fenómenos que se pueden dar en el proceso como son el ruido y silencio documental. Uno de los problemas que surgen en la búsqueda de información es si lo que recuperamos es "mucho o poco", es decir, dependiendo del tipo de búsqueda se pueden recuperar multitud de documentos o simplemente un número muy reducido. A este fenómeno se denomina Silencio o Ruido documental. [7]

- Silencio documental. Son aquellos documentos almacenados en la base de datos pero que no han sido recuperados, debido a que la estrategia de búsqueda ha sido demasiado específica o que las palabras clave utilizadas no son las adecuadas para definir la búsqueda.
- Ruido documental. Son aquellos documentos recuperados por el sistema pero que no son relevantes. Esto suele ocurrir cuando la estrategia de búsqueda se ha definido demasiado genérica.

4.1.1. Concepto de sistema de recuperación de información.

Proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsqueda específicas. Dicha información ha debido ser estructura previamente a su almacenamiento.

Componentes esenciales.

- Documentos estructurados. Es necesario establecer un proceso donde se establezcan herramientas de indización y control terminológico.
- Bases de datos donde estén almacenados los documentos. Definir lenguajes de interrogación y operadores que soportará la base de datos y, establecer que tipo de ecuaciones serán permitidas.

4.1.2. Herramientas.

Lenguajes de indización y control terminológico.

Índices Listado de términos normalizados que representan el contenido de un recurso. Algunos tipos son:

- Índice de materias: términos ordenados según las materias que trata la base de datos, el buscador, etc.
- Índice alfabético: listado de términos alfabéticamente.
- Índice KWIC: Tipo de índice permutado en el que el contenido temático de una obra se representa mediante palabras clave de su título o de otra fuente de información del documento.

Palabras clave (keywords). Término significativo en lenguaje natural que representa el contenido del documento.

En la búsqueda de información esta opción es esencial ya que nos permite acotar y precisar información. El problema recae en definir la palabra exacta que representa el contenido, por ello es conveniente utilizar especificadores. Por ejemplo, si utilizamos la palabra flor en cualquier buscador podemos estar buscando, la floristería más cercana, una imagen de flores o un estudio sobre las flores en las distintas estaciones del año.

- Meta Keywords. La mayoría de los buscadores utilizan para localizar los recursos, las palabras clave de cada página Web.

4.1.3. Técnicas de recuperación de información.

Sistemas de recuperación de lógica difusa.

Esta técnica permite establecer consultas con frases normales, de forma que la máquina al realizar la búsqueda elimina signos de puntuación, artículos, conjunciones, plurales, tiempos verbales, palabras comunes (que suelen aparecer en todos los documentos), dejando sólo aquellas palabras que el sistema considera relevantes. La recuperación se basa en proposiciones lógicas con valores de verdadero y falso, teniendo en cuenta la localización de la palabra en el documento.

Técnicas de ponderación de términos.

Es común que unos criterios en la búsqueda tenga más valor que otros, por tanto la ponderación pretende darle un valor adecuado a la búsqueda dependiendo de los intereses del usuario. Los documentos recuperados se encuentran en función del valor obtenido en la ponderación. El valor depende de los términos pertinentes que contenga el documento y la frecuencia con que se repita. De forma que, el documento más pertinente de búsqueda sería aquel que tenga representado todos los términos de búsqueda y además el que más valor tenga repetidos más veces, independientemente de donde se localice en el documento.

Técnica de clustering

Es un modelo probabilístico que permite las frecuencias de los términos de búsqueda en los documentos recuperados. Se atribuyen unos valores (pesos) que actúan como agentes para agrupar los documentos por orden de importancia mediante algoritmos ranking.

Técnicas de stemming.

Morfológicamente las palabras están estructuradas en prefijos, sufijos y la raíz. La técnica de Stemming lo que pretende es eliminar las posibles confusiones semánticas que se puedan dar en la búsqueda de un concepto, para ello trunca la palabra y busca solo por la raíz.

Algoritmos utilizados para desechar prefijos y sufijos:

- Paice/Husk
- S-stemmer/n-gramas
- Ténicas lingüísticas

Pretenden acotar de una manera eficaz los documentos relevantes. Por esta razón, esta técnica lo consigue mediante una correcta indización en el proceso de tratamiento de los documentos con ayuda de índices, tesauros, etc.; evitando las ambigüedades léxicas y semánticas a la hora de establecer las consultas.

4.2. Similitud Textual.

A continuación se muestra una definición de similitud textual:

Similitud textual es un concepto que se puede ver como una forma de describir la similitud entre secuencias. Una cadena puede contener significado, es decir, la semántica puede ser derivados de ella. La similitud semántica de cadenas es un caso especial de la relación semántica, que tiene sus raíces en la inteligencia artificial y se remonta a 1968.

Similitud semántica se utiliza como una herramienta para encontrar conceptos similares, donde el parecido semántico se utiliza para encontrar los conceptos relacionados. Por ejemplo, un coche y ruedas están más relacionados que un coche y una bicicleta, pero este último sería considerado más similar. [8]

Capítulo 5

Desarrollo del proyecto

5.1. Módulos que integran el proyecto

El proyecto consta de 3 módulos principales (véase la figura 5.1) estos módulos son:

- El módulo extractor de datos.
- El módulo calculador de similitud independiente del idioma.
- El módulo de presentación de resultados.

A continuación explicamos estos puntos:

5.1.1. Módulo extractor de datos.

Este módulo se encarga de extraer datos que serán posteriormente analizados. Se compone de cuatro fases; procesamiento del texto, segmentación del texto, lematización y creación de diccionario. Tiene como entrada tres posibles recursos heterogéneos, las cuales son:

- Base de datos.
- Un par de documentos en formato txt seleccionados por el usuario.
- Un formulario Web con cajas de texto.

Procesamiento del texto

Se obtienen los textos de la entrada elegida y se envían a éste módulo que se encarga de eliminar acentos, caracteres especiales y caracteres de escape, posteriormente todas las letras del texto son cambiadas por minúsculas. La salida de este módulo es un par de textos planos (sin formato, sin etiquetas, etc.). A continuación se muestra un ejemplo para un par de documentos, los textos del lado izquierdo se muestran tal como se obtiene de la entrada y el del lado derecho se muestra como la etapa del módulo los devuelve. De aquí en adelante al primero de estos textos lo llamaremos texto uno y al segundo, texto dos. (véase la figura 5.2)

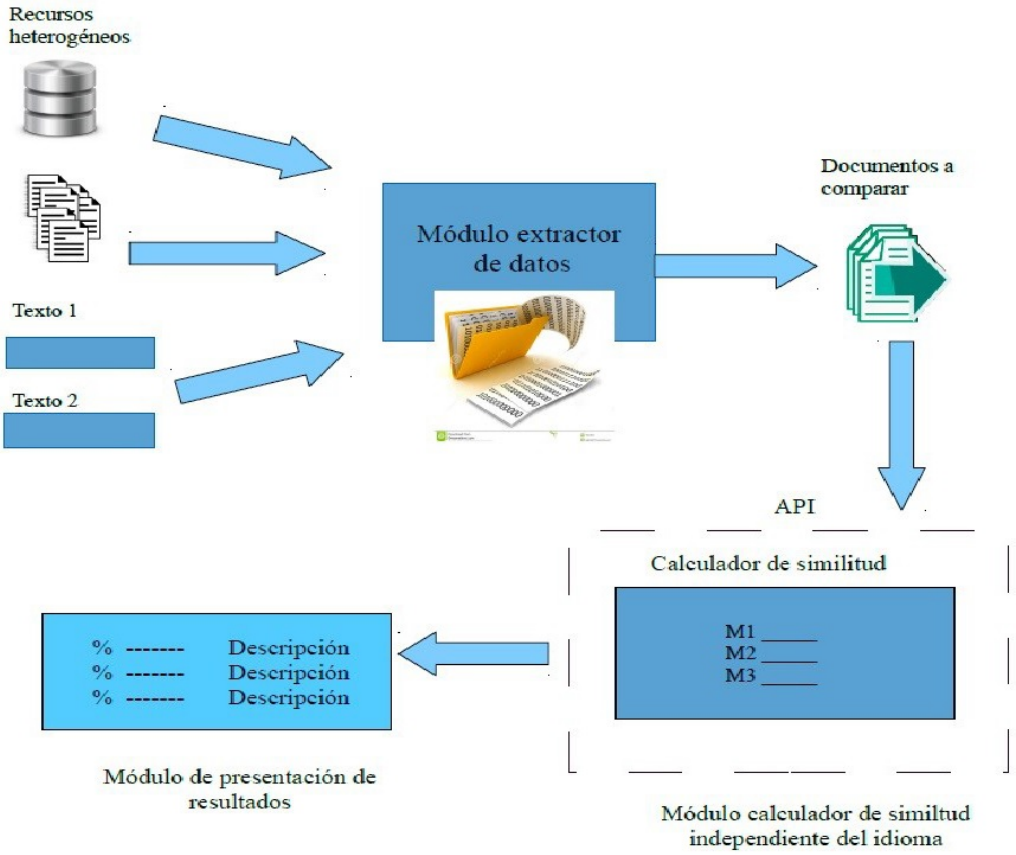

Figura 5.1: Módulos principales del Sistema

Segmentación del texto

En ésta parte del módulo se toman como entrada los textos procesados y se dividen en palabras que se guardan en un arreglo de cadenas ocupando cada palabra una posición del arreglo. A continuación se ejemplifica el resultado de esta operación para el texto uno (véase la tabla 5.1) y para el texto dos (véase la tabla 5.2).

venimos	como	un	socio	hermano	a
finiquitar	los	acuerdos	comerciales	que	expresan
la	voluntad	politica	de	cuatro	pueblos
hermanos	mexico	chile	colombia	y	peru
los	miembros	fundadores	de	la	alianza
del	pacifico				

Tabla 5.1: Arreglo de palabras divididas del texto uno

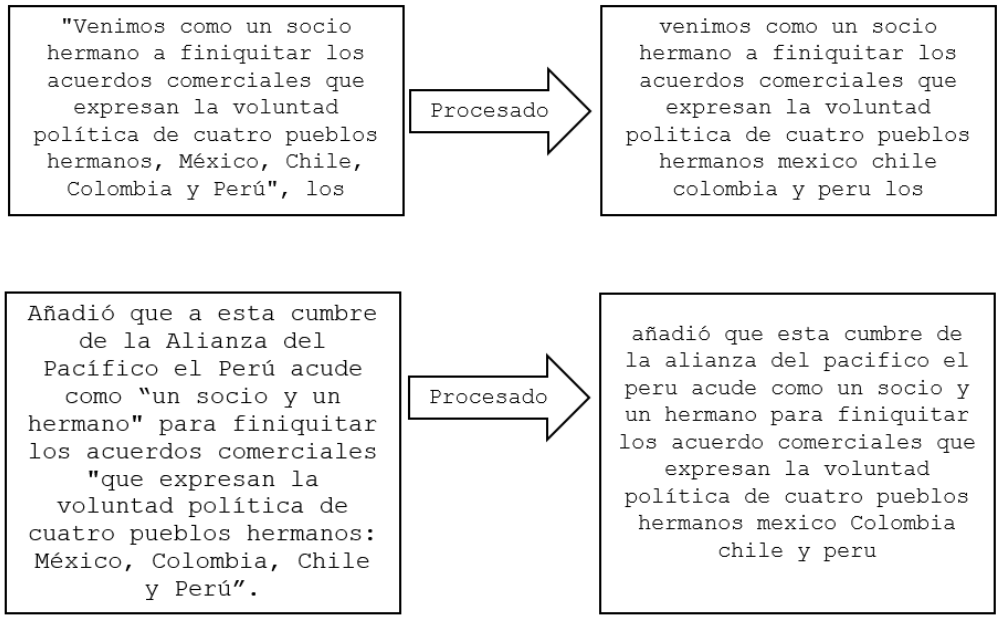

Figura 5.2: Ejemplo de un par de textos antes y después de ser procesados

añadio	que	a	esta	cumbre	de
la	alianza	del	pacifico	el	peru
acude	como	un	socio	y	un
hermano	para	finiquitar	los	acuerdos	comerciales
que	expresan	la	voluntad	politica	de
cuatro	pueblos	hermanos	mexico	colombia	
chile	y	peru			

Tabla 5.2: Arreglo de palabras divididas del texto dos

Lematización

Esta parte del módulo recibe el artículo segmentado por el proceso anterior y mediante el uso de TreTagger [1] se realiza una lematización para obtener los lemas y los almacena en un arreglo de cadenas de donde se recupera posteriormente el lema de cada palabra. En la siguiente tabla se puede ver el resultado de lematizar el vector de palabras obtenido de la división. Es decir, se muestran los lemas de las palabras que conforman el texto uno (veáse la tabla 5.3) seguido de las que conforman el texto dos (veáse la tabla 5.4).

Creación de diccionario

Se toma el arreglo obtenido de la lematización y se obtienen los lemas de cada palabra del primer artículo y se recorre de inicio a fin tomando cada palabra y almacenándola sin

[1] Herramienta para anotar textos con información de part-of-speech y lema, desarrollado en Stuttgart.

venir	como	un	socio	hermanar	a
finiquitar	el	acuerdo	comercial	que	expresar
el	voluntad	politica	de	cuatro	pueblo
hermano	mexico	chile	colombia	y	peru
el	miembro	fundador	de	el	alianza
del	pacificar				

Tabla 5.3: Lista de palabras del texto uno lematizadas

añadio	que	a	este	cumbre	de
el	alianza	del	pacificar	el	peru
acudir	como	un	socio	y	un
hermano	para	finiquitar	el	acuerdo	comercial
que	expresar	el	voluntad	politica	de
cuatro	pueblo	hermano	mexico	colombia	chile
y	peru				

Tabla 5.4: Lista de palabras del texto dos lematizadas

repeticiones en un arreglo de cadenas. De igual forma, se toma el arreglo del segundo artículo y se recorre, si la palabra no está en el diccionario se almacena de lo contrario se pasa a la siguiente palabra. La salida es un arreglo de caracteres que contiene las palabras de los dos artículos a comparar. (véase la tabla 5.5)

venir	como	un	socio	hermanar	a
finiquitar	el	acuerdo	comercial	que	expresar
voluntad	politica	de	cuatro	pueblo	hermano
mexico	chile	colombia	y	peru	miembro
fundador	alianza	del	pacificar	añadio	este
cumbre	acudir	para			

Tabla 5.5: Diccionario de palabras

5.1.2. Módulo calculador de similitud independiente del idioma.

Este módulo se encarga de realizar la comparación de los artículos mediante el uso del diccionario creado y los lemas de las palabras contenidas en cada texto. Para realizar esta tarea se apoyó en la implementación de las métricas binaria y TF que consisten en tomar las palabras del diccionario y compararlas contra las que conforman los artículos.

Métrica Binaria

Si la palabra del diccionario se encuentra en el artículo se agrega un 1 en un vector de enteros que representa verdadero, en caso contrario se agrega un 0 para indicar que

no se encuentra la palabra. Al término de este procedimiento se obtiene un arreglo que contiene enteros(0's o 1's). Siguiendo con el ejemplo, se muestra el contenido de los vectores con el resultado de procesar los textos anteriores. (veáse la figura 5.3) El vector superior corresponde al texto uno y el vector inferior corresponde al texto dos.

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0
0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Figura 5.3: Vectores de comparación por métrica binaria de los artículos contra diccionario.

Métrica TF

Empleamos la métrica Frecuencia del término (TF por sus siglas en inglés) para determinar el número de ocurrencias de una palabra, es decir, las veces que se repite una palabra en un texto indica la relevancia que tiene en el documento. La fórmula 5.1 fue la que se empleó para esta medición :

$$tf(n) = \sum_{n}^{D1} \tag{5.1}$$

Es decir, la frecuencia de aparición de un término (n) en un documento (D1) es la suma de las ocurrencias de dicho término. Por lo tanto, podemos decir que un término que aparece 5 veces tiene más relevancia en el documento que un que aparece sólo una vez. El procedimiento es similar al de la métrica anterior, en un vector de enteros se va agregando el número de ocurrencias en el texto de cada palabra del diccionario.(véase la figura 5.4) El vector superior resulta de aplicar el procedimiento descrito al texto uno, mientras que el inferior es el resultado del texto dos.

1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1
0, 1, 2, 1, 0, 1, 1, 4, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Figura 5.4: Vectores de comparación por métrica TF de los artículos contra diccionario.

En el presente trabajo se toma como entrada el arreglo de cadenas que contiene los lemas de los artículos y el diccionario generado a partir de los mismos. Se toma cada una de las palabras que forman el diccionario, se comparan contra el artículo y se contabiliza las ocurrencias de la palabra. Se almacena en un vector de enteros el número de apariciones de cada palabra.

Similitud del coseno

Una vez representadas las palabras de los artículos como vectores de números podemos medir su similitud. Para ello empleamos el coseno del ángulo entre los vectores como medida de similitud de modo que si se trata de un par de documentos idénticos el ángulo vale 0 y el coseno vale 1. Por otro lado, si son ortogonales el coseno vale 0 indicando que son completamente diferentes.

Por lo tanto, el resultado de la similitud del coseno se encuentra entre el 0 y el 1. En este trabajo se aplica la fórmula de la distancia del coseno a los vectores obtenidos de las métricas binaria y TF para determinar el coseno del ángulo que existe entre dichos vectores. La fórmula se puede ver a en la ecuación 5.2

$$simCoseno = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{5.2}$$

Donde:

- x representa el vector de resultados para el texto uno.
- y representa el vector de resultados para el texto dos.
- n es el la longitud de los vectores.

En la figura 5.5 se muestran los valores que se obtienen para el ejemplo de los textos trabajados anteriormente en base a la métrica binaria. La primer columna contiene las palabras que fueron recopiladas de los textos y almacenadas en el diccionario, en la segunda columna vemos si la palabra del diccionario se encuentra en el primer texto donde 1 indica presencia de la palabra y 0 indica ausencia de la misma, para le tercer columna ocurre lo mismo pero para el segundo texto. Las columnas siguientes contiene el resultado de la multiplicación de X*Y, X*X y Y*Y que son los valores necesarios para el calculo de la distancia del coseno. Al final de la tabla se muestra el total obtenido en cifras, mismas que se introducen en la fórmula del coseno. Específicamente para este par de textos, el resultado que se obtiene del cálculo de similitud por coseno binario es: 0.80714 lo que equivale a 80.71 %.

Diccionario	Texto 1 (X)	Texto 2 (Y)	X*Y	X*X	Y*Y
venir	1	0	0	1	0
como	1	1	1	1	1
un	1	1	1	1	1
socio	1	1	1	1	1
hermanar	1	0	0	1	0
a	1	1	1	1	1
finiquitar	1	1	1	1	1
el	1	1	1	1	1
acuerdo	1	1	1	1	1
comercial	1	1	1	1	1
que	1	1	1	1	1
expresar	1	1	1	1	1
voluntad	1	1	1	1	1
politica	1	1	1	1	1
de	1	1	1	1	1
cuatro	1	1	1	1	1
pueblo	1	1	1	1	1
hermano	1	1	1	1	1
mexico	1	1	1	1	1
chile	1	1	1	1	1
Colombia	1	0	0	1	0
y	1	1	1	1	1
peru	1	1	1	1	1
miembro	1	0	0	1	0
fundador	1	0	0	1	0
alianza	1	1	1	1	1
del	1	1	1	1	1
pacificar	1	1	1	1	1
añadio	0	1	0	0	1
este	0	1	0	0	1
cumbre	0	1	0	0	1
acudir	0	1	0	0	1
un	0	1	0	0	1
para	0	1	0	0	1
colombia	0	1	0	0	1
peru?	0	1	0	0	1
		Total	23	5.291502	5.567764

Figura 5.5: Proceso de archivos mediante métrica binaria

5.1.3. Módulo de presentación de resultados

Este módulo está integrado por clases JSP que conforman la vista del proyecto de modo que un usuario pueda hacer uso del sistema de manera intuitiva. A continuación se presentarán algunas de las ventanas que integran el sistema completo, a manera de ejemplificar su funcionamiento se muestra una corrida con un par de textos en español con la opción de entrada "Seleccionar texto". Primero, se puede observar en la figura 5.6 un menú de opciones de entrada que permite seleccionar el origen de los textos que a continuación serán comparados; la primera opción es desde una base de datos, la segunda opción permite cargar un par de archivos desde un navegador y la tercera permite ingresar directamente los textos.

Para continuar con el ejemplo que se ha venido trabajado, se ha elegido la opción tres y se han cargado los textos en las cajas correspondientes, posteriormente se marcó el idioma español y se mandaron a comparar. En la figura 5.7 se puede ver como ha sido este procedimiento antes de dar click sobre el botón que comienza la comparación.

En la figura 5.8 se muestra una ventana con el resultado de realizar la comparación de los textos mediante las métricas mencionadas y se puede ver que se obtiene una similitud de 80.71 % para la métrica binaria y de 88.21 % para la métrica TF.

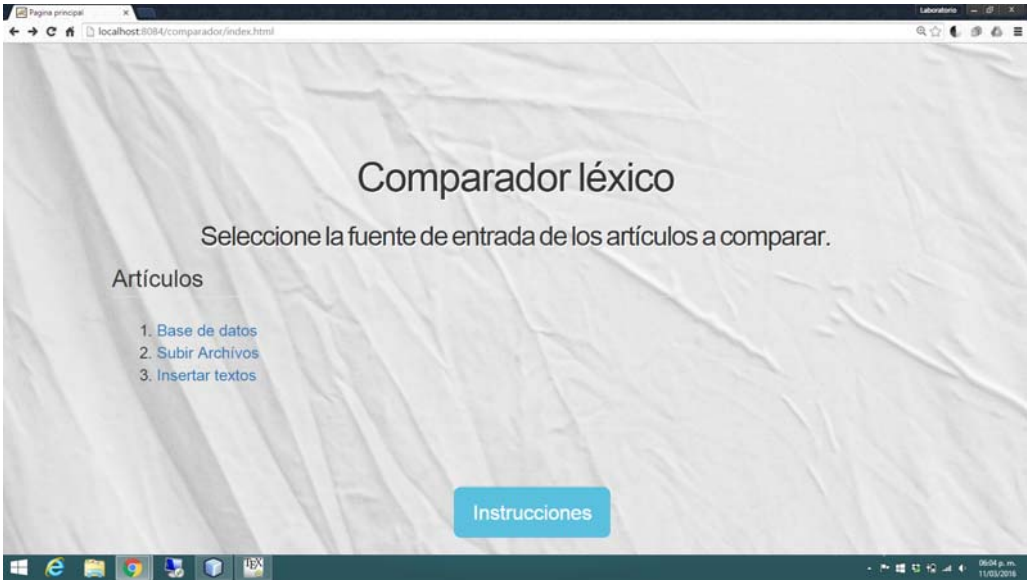

Figura 5.6: Menú de inicio del sistema

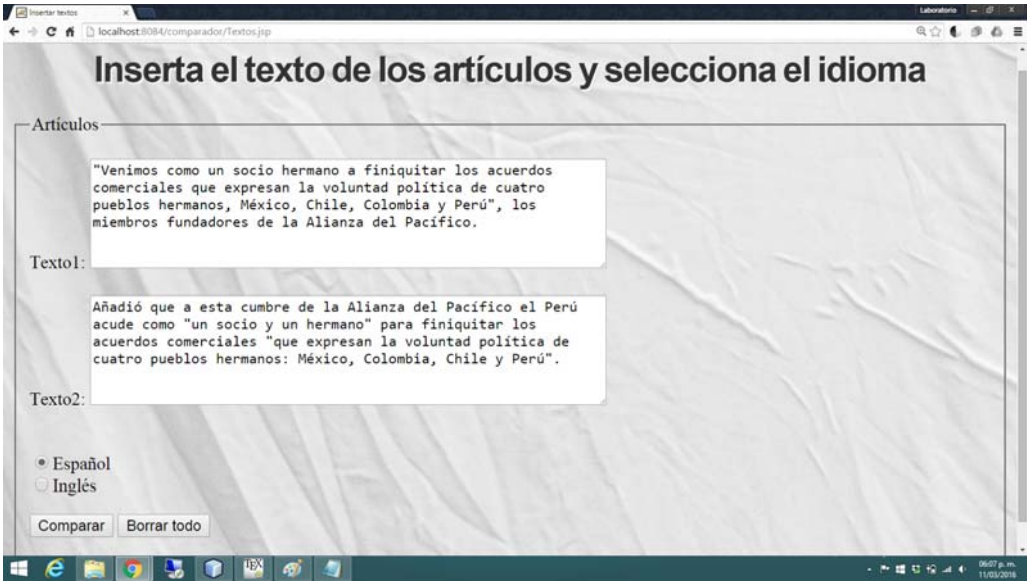

Figura 5.7: Ejemplo de textos cargados en la opción 3 del sistema.

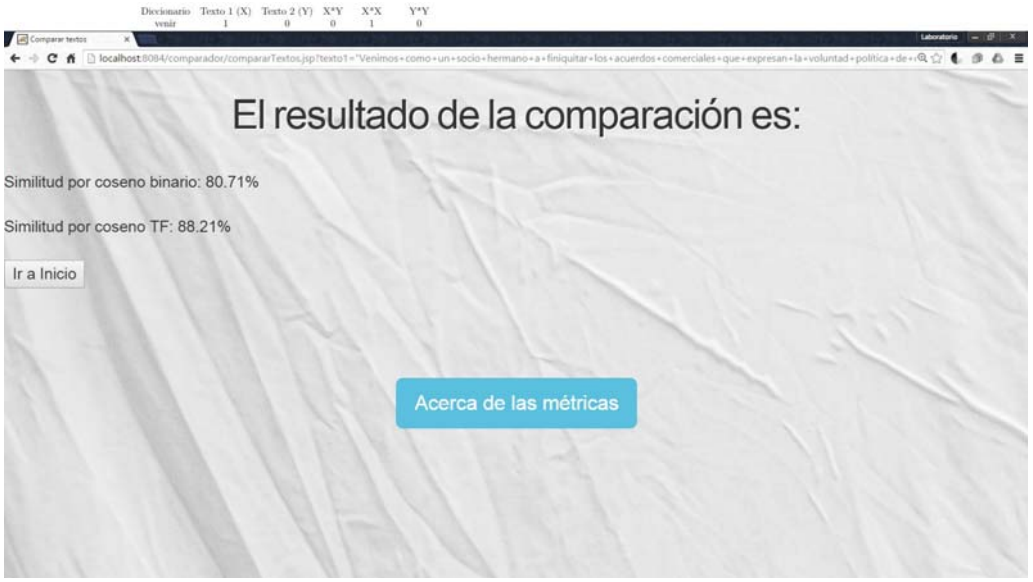

Figura 5.8: Resultado de la comparación de un par de textos en español.

Capítulo 6

Resultados

Se ha realizado el procedimiento para 100 pares de artículos en español y para 100 pares de artículos en inglés. Los artículos han sido recogidos de bases de datos que contienen información de fuentes en Internet y se muestran tal cual se han tomado de dicha base de datos, es decir, no han sido procesados por ninguno de los módulos del programa de modo que se pueden observar caracteres como acentos, comas o comillas entre otros.

Con fines de ejemplificación, se muestra un par de tablas en donde se han vertido los resultados obtenidos de la comparación para 15 pares de artículos en español y 15 pares de artículos en inglés en base a las métricas binaria y TF. La primer columna de la tablas contiene el numero que identifica al par de textos, la segunda columna contiene el texto del primer artículo que será procesado, la tercer columna contiene el texto del segundo artículo que será procesado por el programa. En la cuarta y quinta columna se muestra el resultado de los dos artículos comparados mediante la métrica binaria y métrica TF respectivamente.

6.1. Resultados sobre textos en español

La siguiente tabla contiene los 15 pares de textos en idioma español y el resultado de sus comparaciones.(véase la tabla 6.1)

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
1	El volcamiento de un bus de turismo, ocurrido en el kilómetro 8 de la vía Bogotá-Choachí, dejó un saldo de 38 personas lesionadas, de las cuales 30 tuvieron que ser remitidas a hospitales.	De las personas que iban en el bus y fueron valoradas en el sitio del accidente, 8 no ameritaron traslado a centro asistencial; el resto de los heridos fueron llevados a hospitales.	0.391304	0.645917

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
2	No dejaremos que ninguno de nuestros residentes viva en las sombras", afirmó en español De Blasio durante su discurso sobre el estado de la ciudad, en el que anunció su proyecto.	No dejaremos que ninguno de nuestros residentes viva en la sombra", dijo De Blasio en español al realizar el anuncio durante su discurso del estado de la ciudad, en el que presentó las grandes líneas de su acción de gobierno para 2014.	0.752217	0.900735
3	La nieve y el hielo fueron aparentemente responsables de varios accidentes de tráfico ocurridos en las prefecturas de Ishikawa, Nagano o Aichi que se saldaron en total con 10 fallecidos, informó hoy la cadena pública NHK.	Cuatro de los fallecidos son motoristas, que han perdido la vida en accidentes de tráfico provocados por la nieve y el hielo en las prefecturas de Ishikawa y Nagano.	0.567308	0.789960
4	El presidente de Colombia, Juan Manuel Santos, reconoció hoy las reformas estructurales que su colega Enrique Peña Nieto ha realizado en México y destacó los lazos de amistad que unen a ambas naciones.	El presidente Enrique Peña Nieto reconoció que en el año 2013 la economía mexicana creció por debajo de las expectativas pero aseguró que este años promete un mayor crecimiento económico por las reformas y la inversión extranjera indirecta que alcanzó 35 mil millones de dólares.	0.355579	0.595468

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
5	"Venimos como un socio hermano a finiquitar los acuerdos comerciales que expresan la voluntad política de cuatro pueblos hermanos, México, Chile, Colombia y Perú", los miembros fundadores de la Alianza del Pacífico.	Añadió que a esta cumbre de la Alianza del Pacífico el Perú acude como ?un socio y un hermano"para finiquitar los acuerdos comerciales "que expresan la voluntad política de cuatro pueblos hermanos: México, Colombia, Chile y Perú?.	0.780670	0.873540
6	Solís, un historiador de 55 años poco conocido en el país cuando inició la campaña política, ascendió vertiginosamente en dos semanas, superando también al candidato del izquierdista Frente Amplio , José María Villalta, a quien todas los sondeos daban como el más seguro rival de Araya en una segunda vuelta.	Solís, un historiador de 55 años poco conocido en el país cuando inició la campaña presidencial, tuvo un ascenso vertiginoso en menos de tres semanas, sobrepasando también al izquierdista José María Villalta, a quien todos daban como el casi seguro rival de Araya en una segunda vuelta.	0.760041	0.842857
7	El asesino, que supuestamente trabajaba como guardia de seguridad, ha sido detenido por la policía local.	Según RIA Novosti, el agresor es un joven de 25 años de edad que trabajaba en una agencia de seguridad y ya se encuentra detenido.	0.394405	0.416666

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
8	La querella del Comité de Apoyo al Tibet por la cual se inició la investigación y que ahora es reproducida por el juez en el auto para decretar la prisión del expresidente, asegura que "Jiang Zemin ejerció autoridad de supervisión sobre las personas que cometieron de forma directa los abusos propiamente dichos, lo que le hace responsable de actos de tortura y otros importantes abusos de derechos humanos perpetrados por sus subordinados contra la población tibetana".	"Jiang Zemin ejerció autoridad de supervisión sobre las personas que cometieron de forma directa los abusos propiamente dichos, lo que le hace responsable de actos de tortura y otros importantes abusos de derechos humanos perpetrados por sus subordinados contra la población tibetana", señala la querella que ahora el juez reproduce en el auto para decretar la prisión del expresidente.	0.782584	0.958450
9	Once personas murieron, más de 1.000 resultaron heridas y decenas de miles quedaron sin electricidad cuando la peor tormenta de nieve en décadas afectó Tokio y sus alrededores antes de dirigirse hacia al norte, a la costa del Pacífico afectada por el tsunami en 2011.	Tokio, vivió la mayor nevada en 20 años con 27 centímetros de nieve acumulada.	0.243975	0.387940
10	Este activista ucraniano muy activo dentro de la oposición contra el presidente Viktor Yanukovitch afirma haber sido secuestrado el 22 de enero en Kiev y torturado durante una semana, antes de ser liberado en medio de un bosque.	Este opositor de 35 años y padre de tres niños fue secuestrado el 22 de enero en Kiev y torturado durante una semana antes de ser liberado en medio de un bosque.	0.633446	0.790167

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
11	Su identidad la había revelado durante el viaje en patrullera desde Ebon, dijo un funcionario del ministerio de Relaciones Exteriores de las Marshall, un vasto archipiélago de Micronesia.	La información fue suministrada por el pescador al embajador mexicano en Manila, Julio Camarena, quien también habló del caso con el ministro de Exteriores de las Islas Marshall, Phillip Muller, así como otras autoridades del archipiélago.	0.224701	0.523682
12	Estados Unidos informó este domingo que Corea del Norte retiró la invitación para que un enviado especial visitara Pyongyang, con el objetivo de negociar la liberación de Kenneth Bae, el ciudadano estadounidense condenado a 15 años de trabajo forzado.	Agencia EFE ı Febrero 10 de 2014 Corea del Norte Washington, 10 feb .- Corea del Norte canceló este domingo su invitación para que el enviado de Washington Robert King visitase Pyongyang con el fin de hablar sobre el ciudadano estadounidense Kenneth Bae, preso en el país desde 2012, informó el Departamento de Estado a la cadena CNN.	0.552157	0.750404
13	Desde el comienzo de las protestas sociales del miércoles pasado, han dimitido en Bosnia los Gobiernos de tres de los diez cantones que forman la Federación croata-musulmana, así como el director del cuerpo de coordinación de la Policía bosnia.	En Bihac, en el extremo oeste, donde el sábado hubo unos choques entre los manifestantes y la policía, unas 500 personas se volvieron a congregar ayer en la principal plaza para exigir la dimisión del ministro cantonal, quien según algunos medios se ha fugado del país.	0.162050	0.535371

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
14	El responsable de la OIEA, Ali Akbar Salehi, afirmó esta semana que Irán estaría dispuesto a realizar .ªlgunas modificaciones en los planos para producir menos plutonio", si bien insistió que se trata de un reactor de investigación.	En respuesta a esas inquietudes, el jefe de la OIEA, Ali Akbar Salehi, afirmó esta semana que Irán está dispuesto a "hacer algunas modificaciones en los planes para producir menos plutonio", aunque repitió que Arak es un reactor de investigación.	0.707776	0.785905
15	La primera vuelta de las elecciones presidenciales de Colombia se realizará el 25 de mayo y en caso de que ninguno de los candidatos obtenga más de un 50 por ciento de los votos, los dos primeros se enfrentarán a fines de junio en una segunda ronda.	En el caso de que Santos enfrente a Zuluaga en el balotaje, el actual presidente obtendría un 38 por ciento de los votos, frente a un 18 por ciento de su rival.	0.415227	0.723922

Tabla 6.1: Resultado de la comparación de artículos en español.

6.2. Resultados obtenidos sobre textos en inglés

La tabla siguiente muestra los resultados obtenidos de las pruebas que se realizaron con 15 pares de textos en inglés, las últimas dos columnas corresponden a los resultados obtenidos de la medición basada en las métricas mencionadas. (véase la tabla 6.2)

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
1	Mall attackers used 'less is more' strategy.	In Kenya, attackers used 'less is more'	0.714285	0.714285
2	Strategy opposition leaders emerge to commemorate Cambodian workers' deaths.	Çambodia opposition leaders summoned to court Weak earnings drag stocks lower. "	0.301511	0.301511

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
3	8 quake strikes off Solomon Islands.	Magnitude 6.3 quake strikes off Solomon Islands.	0.912870	0.912870
4	mexico wishes to guarantee citizens' safety.	mexico wishes to avoid more violence.	0.5	0.5
5	spain currently holds the rotating presidency of the osce.	spain currently holds the osce's presidency.	0.801783	0.797724
6	the treaty was first signed in 1990.	the cfe treaty was signed in 1990.	0.833333	0.833333
7	gorgich and pashtoon were executed in prison.	gorgich and pashtoon were executed for trafficking heroin in sistan-baluchestan province.	0.654653	0.654653
8	safe bourada was sentenced to 15 years in prison.	djamel badaoui was sentenced to five years.	0.534522	0.534522
9	santos stated the radioactive material was the primary basis for generating crude weapons of mass destruction and terrorism.	uranium is a radioactive material which is the primary basis for generating dirty weapons of mass destruction and terrorism.	0.800326	0.780200
10	fulvio berghella stated that the digital worm named sq hell was particularly virulent and replicated itself at the rate of 8000 times an hour.	fulvio berghella stated that the digital worm named sq hell has a very high capacity to replicate itself and the digital worm named sq hell is slowing down the poste italiane computer network and making some computers inoperable.	0.554265	0.656532
11	A cat standing on tree branches.	A black and white cat is high up on tree branches.	0.615457	0.615457
12	Two green and white trains sitting on the tracks.	Two green and white trains on tracks.	0.881917	0.881917
13	A small white cat with glowing eyes standing underneath a chair.	A white cat stands on the floor.	0.478091	0.524142

Continua en la página siguiente.

Número	Texto artículo uno	Texto artículo dos	Binaria	TF
14	A large boat in the water at the marina.	A large boat on the sea.	0.577350	0.615457
15	a bus driving in a street.	Red double decker bus driving down street.	0.507092	0.400891

Tabla 6.2: Resultado de la comparación de pares de artículos en inglés.

Capítulo 7

Análisis y discusión de resultados

El procedimiento se aplicó sobre artículos recogidos de documentos en Internet y que están contenidos en bases de datos. Los resultados que se obtuvieron sobre los pares de textos comparados demuestran que la longitud del texto influye bastante en los resultados, es decir, entre más palabras contiene el texto, el vector resulta más largo permitiendo calcular la distancia del coseno con más elementos, este es el caso para la mayoría de textos en español mientras que los textos analizados en inglés son textos más cortos que producen vectores más pequeños.

Análisis sobre textos en español. En la tabla de textos en español se observa que la similitud binaria y la similitud TF generalmente tienen diferencia significativa, siendo la métrica TF la que arroja resultados más altos. Por otro lado, la mayoría de estos textos contienen palabras similares entre ellos, un caso particular es el par número 13 que contiene a propósito textos sobre temas diferentes y se observa que la similitud es de un 16.20 % para la binaria y de un 53.53 % para la TF. Esto se debe a que el lema de algunas palabras es el mismo y al compararse obtenemos similitud sin que necesariamente los textos traten un mismo tema.

En cambio, si tomamos como referencia el par número 6 que contiene textos muy parecidos vemos que los resultados de la similitud son altos 76 % para la métrica binaria y 84.28 % para la TF.

Análisis sobre textos en inglés. En la tabla que contiene los resultados de los textos en inglés encontramos que los resultados obtenidos con ambas métricas son muy similares y en muchos casos idénticos, esto se debe a que el análisis se hizo sobre textos cortos y con palabras similares produciendo vectores cortos y muy parecidos entre si. Aquí podemos observar que el resultado para TF no siempre es mayor, por ejemplo en el par 9 se tiene un porcentaje de 80.03 % para binaria y de 78.02 % para TF. De manera similar, los resultados mostrados en el par número 15 reflejan un porcentaje de similitud binaria mayor a TF aproximadamente en un 10 %.

Capítulo 8

Conclusiones

Al término de este proyecto de integración podemos concluir que:

- El sistema desarrollado permite calcular la similitud textual entre un par de textos que puede estar en idioma inglés o español, obtenidos a partir de tres recursos diferentes, aplicando métricas y usando la distancia del coseno, por lo cual se alcanzó el objetivo buscado.
- Se obtuvo resultados más acertados en textos de longitudes mayores como fue el caso de los textos en español que contienen cantidades de palabras más grandes que los textos en inglés. Por otro lado, se observó que aunque los textos no tengan relación semántica, se obtiene un porcentaje de similitud debido a que algunas palabras con el mismo lema se pueden encontrar en los dos textos y al realizar la comparación se obtiene un resultado.
- En la parte de la lematización se encontró que algunas palabras no correspondían con el lema que se le asignaba, esto se debe a la herramienta empleada que no identificaba correctamente la relación, pero a pesar de esto obtuvimos resultados aceptables al emplear el sistema para n-gramas igual a 1.
- Los resultados de este proyecto de integración pueden ser mejorado a futuro con la complementación de n-gramas mayores que permitan relacionar las palabras que integran el texto y determinar la similitud que existe entre los ambos documentos.

Apéndice A

A continuación se presenta el código de las principales clases de la aplicación en secciones por paquetes y las clases contenidas en ellos.

A.1. Paquete manipularArchivos

Clase Controlador

```
\texttt{package manipularArchivos;

import Procesos.*;
import java.io.IOException;
import java.util.ArrayList;
import mx.uam.azc.ws.posTag.Token;
import mx.uam.azc.ws.posTag.TreeTagger;
import org.annolab.tt4j.TreeTaggerException;
}
/**
 *
 * @author Erik
 */
public class Controlador {

    float sCoseno=0.5f;
    double cosenob;
    double cosenotf;
    ArrayList entrada = new ArrayList();

    public Controlador() {
    }



    public float Ejecutar(ArrayList entrada, String idioma)
            throws IOException, TreeTaggerException {
        ArrayList<String> d = new ArrayList<>();
        Procesador pro = new Procesador();

        //DECLARACION DE VARIABLES A USAR
        String enPalabras[], enPalabras2[];
        Comparador comp = new Comparador();
```

```
ArrayList<Token> tokens = new ArrayList<>();
ArrayList contA1 = new ArrayList<>();
ArrayList contA2 = new ArrayList<>();
ArrayList vecesA1 = new ArrayList<>();
ArrayList vecesA2 = new ArrayList<>();
ArrayList<String> lemas1 = new ArrayList<>();
ArrayList<String> lemas2 = new ArrayList<>();
String proceso1 = "", proceso2 = "";

//RECUPERA LOS DOS ARTICULOS A COMPARAR
String entrada1 = (String) entrada.get(0);
String entrada2 = (String) entrada.get(1);
//pasar a metodo lector
System.out.println("Archivos␣sin␣procesar:␣");
System.out.println(entrada1);
System.out.println(entrada2);

System.out.println("Archivos␣procesados:␣");
proceso1 = pro.preprocesarTexto(entrada1);
proceso2 = pro.preprocesarTexto(entrada2);

System.out.println(proceso1);
System.out.println(proceso2);

//DIVIDE LA CADENA EN PALABRAS
enPalabras = pro.dividir(proceso1);
enPalabras2 = pro.dividir(proceso2);


TreeTagger unTagger = new TreeTagger();
tokens = unTagger.Etiquetar(enPalabras, idioma);


for (Token aux : tokens) {
    lemas1.add(aux.getLema());
}

tokens = unTagger.Etiquetar(enPalabras2, idioma);

for (Token aux : tokens) {
    lemas2.add(aux.getLema());
}
System.out.println("\n\nResultado␣"
        + "del␣etiquetado␣art1:␣");
for(String cont: lemas1)
{
    System.out.println(cont);
}

System.out.println("\n\nResultado␣"
        + "del␣etiquetado␣art2:␣");
for(String cont: lemas2)
{
    System.out.println(cont);
}
```

```
        System.out.println("\n\nContenido␣del␣diccionario:␣");
        //CREAR EL DICCIONARIO
        d = pro.crearD(lemas1, lemas2);
        for (String contenido : d) {
            System.out.println(contenido);

        }

        //COMPARAR EL ARTICULO1 CONTRA DICCIONARIO
        contA1 = comp.calcularBin(d, lemas1);
        vecesA1 = comp.calcularTF(d, lemas1);

        System.out.println("\nEl␣resultado"
                + "␣de␣la␣comparacion␣n-gramas=1"
                    + "␣de␣articulo␣1␣es:␣\n");

        System.out.println("BINARIO\n" + contA1);
        System.out.println("TF\n" + vecesA1);

        //COMPARAR EL ARTICULO2 CONTRA DICCIONARIO
        contA2 = comp.calcularBin(d, lemas2);
        vecesA2 = comp.calcularTF(d, lemas2);

        System.out.println("\nEl␣resultado␣"
                + "de␣la␣comparacion␣n-gramas=1␣"
                    + "de␣articulo␣2␣es:␣\n");

        System.out.println("BINARO\n" + contA2);
        System.out.println("TF\n" + vecesA2);

        //MEDIR SIMILITUD DEL COSENO
        sCoseno = comp.simiCoseno(contA1, contA2);
        this.cosenob = comp.simiCoseno(contA1, contA2);
        this.cosenotf = comp.simiCoseno(vecesA1, vecesA2);
        return sCoseno;
    }

    public float getsCoseno() {
        return sCoseno;
    }

    public double getCosenob() {
        return cosenob;
    }

    public double getCosenotf() {
        return cosenotf;
    }

}
```

A.2. Paquete Procesos

Clase Comparador

```
package Procesos;

import java.io.IOException;
import static java.lang.Math.*;
import java.util.ArrayList;
import org.annolab.tt4j.TreeTaggerException;

/**
 *
 * @author Erik
 */
public class Comparador {

  public ArrayList<Integer> calcularBin(
              ArrayList<String> diccionario,
                 ArrayList<String> articulo ) throws
                      IOException, TreeTaggerException{

        ArrayList pesos = new ArrayList<>();

        for(String palabra:diccionario){

           if(articulo.contains(palabra))
             pesos.add(1);
           else
             pesos.add(0);
        }
        return pesos;
    }

  public ArrayList<Integer> calcularTF(
      ArrayList<String> diccionario,ArrayList<String>articulo)
          throws IOException, TreeTaggerException{

        int contador;
        ArrayList<Integer> pesos = new ArrayList<>();

        for(String palabra:diccionario){
            contador=0;
            for(String pal:articulo){
                if(palabra.equalsIgnoreCase(pal)){
                    contador++;
                }
            }

           pesos.add(contador);
        }
        return pesos;
    }
```

```
        //cálculo la distancia del coseno
       public float simiCoseno(ArrayList x, ArrayList y)
       {
           int sumas=0;
           int sumax=0;
           int sumay=0;
           float coseno=0.0f;

        for (int i = 0; i <= x.size()-1; i++) {
               Integer  a=Integer.parseInt(x.get(i).toString());
               Integer  b=Integer.parseInt(y.get(i).toString());
               sumas+=(a*b);

           }
        for (int i = 0; i <=x.size()-1; i++) {
           sumax+=pow(Integer.parseInt(x.get(i).toString()),2);
           }
        for (int i = 0; i<=x.size()-1; i++) {
           sumay+=pow(Integer.parseInt(y.get(i).toString()),2);
           }

         coseno=(float) (sumas/((sqrt(sumax))*(sqrt(sumay))));
           return coseno;
       }

}
```

Clase Procesador

```
package Procesos;

import java.util.ArrayList;

/**
 *
 * @author Erik
 */
public class Procesador {
    String [] r ;
    public Procesador() {
    }


    ArrayList<String> diccionario=new ArrayList<>();

    public ArrayList<String> crearD(ArrayList<String>
            entrada1, ArrayList<String> entrada2)
    {
        //analizar entrada1
        for(String tmp:entrada1){

           if(diccionario.contains(tmp))
```

```
            continue;
          else
            diccionario.add(tmp);
        }
        //analizar entrada2
        for(String tmp:entrada2){

          if(diccionario.contains(tmp))
            continue;
          else
            diccionario.add(tmp);
        }
        return diccionario;
    }


     public String preprocesarTexto(String texto) {
          String cadena="";
          texto=texto.replaceAll("\n", "␣");
          texto=texto.replaceAll("[,.]", "␣");
          texto=texto.replaceAll("[\\'\\(\\)]", "␣");
          texto=texto.replaceAll("\\d", "");
          texto=texto.replaceAll("␣␣", "␣");
          texto=texto.replaceAll("[;:]", "␣");
          texto=texto.replaceAll("\"","");
          texto=texto.replaceAll("[¿?¡!]","");
          texto=texto.replaceAll("á", "a");
          texto=texto.replaceAll("é", "e");
          texto=texto.replaceAll("í", "i");
          texto=texto.replaceAll("ó", "o");
          texto=texto.replaceAll("ú", "u");
          //texto=texto.replaceAll("ñ", "n");
          cadena= texto.replaceAll("(?<=\\p{Ll})"
         + "(?=\\p{Lu})|(?<=\\p{L})(?=\\p{Lu}\\p{Ll})", "␣");
          cadena= cadena.replaceAll("[-_]", "␣");

          //pasa la cadena procesada a minusculas
          cadena=cadena.toLowerCase();

          return cadena;
    }

     public String[] dividir(String texto){
         r = texto.split("␣");
         return r;
     }
}
```

A.3. Paquete conexionDB

Clase ConexionDB

```
package conexionDB;

import java.sql.*;
import java.io.IOException;
import java.util.ArrayList;
import manipularArchivos.Controlador;
import org.annolab.tt4j.TreeTaggerException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
/**
 *
 * @author Erik
 */
public class ConexionDB {

    private static Connection conexion;
    Controlador con=new Controlador();
    double coseno, cosb, costf;
    ArrayList entrada=new ArrayList();
    String idPar;
    int id;

    String BD="academia1";
    String usuario="root";
    String contraseña="labsim";

    public void Conectar() throws
            IOException, TreeTaggerException{
        try {
            // Cargar el driver
            Class.forName("com.mysql.jdbc.Driver");
            // Se obtiene una conexión con la base de datos
   conexion = DriverManager.getConnection("jdbc:mysql://"
                  + "localhost:3306/"
                          +BD, usuario, contraseña);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ConexionDB.class.getName()).
                    log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(ConexionDB.class.getName()).
                    log(Level.SEVERE, null, ex);
        }
    }

    public void cerrarConexion() {
        try {
            conexion.close();
            System.out.println("Se␣ha␣finalizado␣la␣"
                            + "conexión␣con␣el␣servidor");
        } catch (SQLException ex) {
            Logger.getLogger(ConexionDB.class.getName()).
                    log(Level.SEVERE, null, ex);
        }
```

```
    }

     public void leerDato(String idioma)
             throws IOException, TreeTaggerException {
        try {
         String query = "SELECT * FROM Articulos "
                 + "WHERE Idioma='"+idioma+"'";
            Statement st = conexion.createStatement();
            ResultSet rs;
            rs = st.executeQuery(query);
            rs.beforeFirst();
            while (rs.next()) {
                //imprimir los registros de la BD
                String ar1=rs.getString("Articulo1");
                String ar2=rs.getString("Articulo2");

                entrada.add(0,ar1);
                entrada.add(1,ar2);
                idPar=rs.getString("idArticulo");
                id= Integer.parseInt(idPar);
                con.Ejecutar(entrada, idioma);
                insertarDato(id);
                cosb = con.getCosenob();
                costf = con.getCosenotf();
                System.out.println("\n\n Similitud por "
                        + "cosenoBIN: "+cosb);
                System.out.println("Similitud por "
                        + "cosenoTF: "+costf+"\n\n");
            }

        } catch (SQLException ex) {
            System.out.println(ex.getCause());
            JOptionPane.showMessageDialog(null,
                    "Error en la adquisición de datos");
        }
    }

    public void insertarDato(int id) {
        try {
            //actualizar datos en la Base
            String query = "UPDATE articulos SET "
                    + "SimilitudBIN='"+cosb
                     +"', SimilutudTF='"+costf+
                        "' WHERE idArticulo="+(id-1);
            PreparedStatement stmt =
                    conexion.prepareStatement(query);
            stmt.executeUpdate(query);
            System.out.println("Datos almacenados"
                            + " de forma exitosa");
        } catch (SQLException ex) {
            System.out.println(ex.getCause());
            JOptionPane.showMessageDialog(null, "Error en el "
                             + "almacenamiento de datos");
        }
    }
```

A.4. Paquete mx.uam.azc.ws.posTag

Clase Token

```
package mx.uam.azc.ws.posTag;

/**
 *
 * @author: Proporcionada por el Profesor
 */

public class Token {
   private String  palabra;
    private String categoria;
    private String lema;


    public Token(String palabra, String categoria,
                                String lema) {
        this.palabra = palabra;
        this.categoria = categoria;
        this.lema = lema;
    }

    public Token() {
    }


    public String getPalabra() {
        return palabra;
    }

    public void setPalabra(String palabra) {
        this.palabra = palabra;
    }

    public String getCategoria() {
        return categoria;
    }

    public void setCategoria(String categoria) {
        this.categoria = categoria;
    }

    public String getLema() {
        return lema;
    }

    public void setLema(String lema) {
        this.lema = lema;
    }

    @Override
```

```
    public String toString() {
        return "Token{" + "palabra=" + palabra +
                ",␣categoria=" + categoria +
                        ",␣lema=" + lema + '}';
    }

}
```

Clase TreeTagger

```
package mx.uam.azc.ws.posTag;


import java.io.IOException;
import java.util.ArrayList;
import static java.util.Arrays.asList;
import org.annolab.tt4j.TokenHandler;
import org.annolab.tt4j.TreeTaggerException;
import org.annolab.tt4j.TreeTaggerWrapper;

/**
 *
 * @author: Proporcionada por el Profesor
 */
public class TreeTagger {

    public ArrayList<Token> Etiquetar(
            String [] cadenaEntrada, String idioma)
                 throws IOException, TreeTaggerException{
        final ArrayList<Token> tokens = new ArrayList<Token>();

      System.setProperty("treetagger.home", "C:/TreeTagger");

        TreeTaggerWrapper<String> tt =
                new TreeTaggerWrapper<String>();
        try {

         //PRUEBA SI LA ENTRADA ES EN ESPAÑOL O EN INGLES
          if("ES".equals(idioma)){
            tt.setModel(
                "C:/TreeTagger/modelos/spanish.par:iso8859-1");
                }
          else
            tt.setModel(
                "C:/TreeTagger/modelos/english.par:iso8859-1");
             tt.setHandler(new TokenHandler<String>() {
                public void token(
                    String token, String pos, String lemma) {
                      tokens.add(new Token(token,pos,lemma));

                        }
                });
```

```
                tt.process(asList(cadenaEntrada));
            }
            finally {
                tt.destroy();
            }
        return tokens;
    }
}
```

Apéndice B

En este apéndice se muestran las clases que integran el bloque de presentación.

B.1. Páginas Web

B.1.1. index

```
<html>

    <head>
        <title>Pagina principal</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
            scale=1.0">
        <link rel="stylesheet" type="text/css" href="Estilos.css">

        <div class="container">

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap
      /3.3.6/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
      jquery.min.js"></script>
  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
      bootstrap.min.js"></script>



    </head>

    <body>

        <h1>Comparador léxico</h1>

              <div></div>


        <h3>Seleccione la fuente de entrada de los artículos a comparar
            .</h3>
         <fieldset>
```

```
        <legend> Artículos </legend>
        <ol>
         <li> <a href="bases.jsp">  Base de datos  </a></li>
         <li> <a href="cargar.jsp"> Subir Archívos   </a></li>
         <li> <a href="Textos.jsp"> Insertar textos   </a></li>
         </ol>

         </fieldset>

<div id="inst" class="container">

<!-- Trigger the modal with a button -->
<center>

<button type="button" class="btn btn-info btn-lg" data-toggle="modal"
    data-target="#myModal">Instrucciones</button>
</center>
<!-- Modal -->
<div class="modal fade" id="myModal" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&
              times;</button>
          <h4 class="modal-title">Instrucciones de uso.</h4>
      </div>
      <div class="modal-body">
        <p><UL type = disk >

        </p>

        <ul>
            <li type="square">COMPARAR BASES DE DATOS
                <ol>
                    <li>Seleccionar la opción <strong>Bases de datos<
                        /strong> para comparar el contenido en una
                        base de datos. El
                         contenido de la base de datos son pares de
                             artículos en idioma inglés o español.</li
                            >
                    <li>En la página que se muestra deberá
                        seleccionar el idioma en el que se encuentran
                         lo textos.</li>
                </ol>
            </li>
            <br />
            <li type="square">SUBIR ARCHIVOS
                <ol>
                    <li>Seleccionar la opción de <strong>Subir
                        archívos</strong> para cargar un par de
                        textos en formato .txt.
                    </li>
```

```
                    <li>En la página que aparece, dé click sobre los
                        botones de selección y en un explorador de
                        archívos deberá
                         elegir el archívo que desea cagar. Seleccione
                             el idioma del contenido de los artículos
                            .</li>
                </ol>
            </li>
            <br />
            <li type="square">INSERTAR TEXTOS
                <ol>
                    <li>Seleccionar la opción de <strong>Insertar
                        textos</strong> para escribir texto dentro de
                         cajas .
                    </li>
                    <li>En la página que aparece, deberá escribir el
                        texto que desea en las cajas estiquetadas
                        como
                         Texto 1 y Texto 2. Seleccione el idioma que
                             corresponde al texto introducido.</li>
                </ol>
            </li>
            <br />

            <li type="square">COMPARAR
                <ol>
                    <li>Para comenzar el proceso de comparación debe
                        dar click sobre el boton <strong>Comparar</
                        strong> y comenzará
                         a procesar el contenido de la base de datos.<
                            /li>

                </ol>
            </li>
            <br />
            <li type="square">BORRAR TODO
                <ol>
                    <li>Si necesita corregir los datos, puede dar
                        click en el botón <strong>Borrar todo</strong
                        > para limpiar
                         todos los datos ingresados así como la
                             casilla seleccionada de idioma</li>
                </ol>
            </li>
        </ul>

      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="
            modal">Cerrar</button>
      </div>
    </div>

  </div>
</div>
```

```
</div>


    </body>
</html>
```

B.1.2. Textos

```
<%--
    Document   : Textos
    Created on : 6/01/2016, 06:32:10 PM
    Author     : Erik
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>Insertar textos</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">
    </head>
    <body>
        <h1>Inserta el texto de los artículos y selecciona el idioma</
            h1>

        <form action="compararTextos.jsp" method="get">
        <fieldset>
        <legend> Artículos </legend>

        <p><label for="texto1">Texto1:</label>
        <textarea name="texto1" id="texto1" rows="6" cols="60"></
            textarea>
        </p>

        <p><label for="texto2">Texto2:</label>
        <textarea name="texto2" id="texto2" rows="6" cols="60"></
            textarea>
        </p>

        <br>

        <input type="radio" name="idioma" value="ES"/>Español<br>
        <input type="radio" name="idioma" value="IN"/>Inglés<br>

        <p>
        <input type="submit" value="Comparar" />
        <input type="Reset" value="Borrar todo"/>
        </p>
         </fieldset>
        </form>
```

```
    </body>
</html>
```

B.1.3. bases

```
<%--
    Document   : bases
    Created on : 5/01/2016, 10:28:43 PM
    Author     : Erik
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>Seleccionar BD</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">
    </head>
    <body>
        <h1>Seleccione el idioma de los textos para comenzar el proceso
            </h1>
              <form action="compararBD.jsp" method="get">
        <fieldset>
        <legend> Base de datos </legend>

        <p><label>Se aplicará el proceso sobre la base de datos
            Academia1</label>

        </p>

        <input type="radio" name="idioma" value="ES"/>Español<br>
        <input type="radio" name="idioma" value="IN"/>Inglés<br>

        <p>
        <input type="submit" value="Comparar" />
        <input type="Reset" value="Borrar todo"/>
        </p>
         </fieldset>
        </form>
    </body>
</html>
```

B.1.4. cargar

```
<%--
    Document   : cargar
    Created on : 5/01/2016, 06:07:49 PM
    Author     : Erik
```

```
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>cargar</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">
    </head>
    <body>
        <h1>Selecciona el archivo a cargar</h1>

        <form action="compararArchivos.jsp" method="post" enctype='
            multipart/form-data' >
        <fieldset>
        <legend> Cargar dos archivos .txt para ser comparados </legend>

        <p>

             <label for="file1">Archivo1:   <input type="file"  name="
                 file1"  />
             </p>


           <p>
             <label for="file2">Archivo2:   <input type="file" name="
                 file2"  />

             </p>

        <input type="radio" name="idioma" value="ES"/>Español<br>
        <input type="radio" name="idioma" value="IN"/>Inglés<br><br><br
            >


                <input type="submit" value="Comparar" />
                <input type="Reset" value="Borrar todo"/>
 </fieldset>
</form>

    </body>
</html>
```

B.1.5. compararArchivos

```
<%--
    Document   : compararArchivos
    Created on : 8/01/2016, 05:08:00 PM
    Author     : Erik
--%>
<%@page import="java.text.DecimalFormat"%>
```

```
<%@page import="java.io.IOException"%>
<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="java.io.File"%>
<%@page import="Procesos.MiServlet"%>
<%@page import="java.util.ArrayList"%>
<%@page import="manipularArchivos.Controlador"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.ArrayList"

         errorPage=""%>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>Comparar Archivos</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap
      /3.3.6/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
      jquery.min.js"></script>
  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
      bootstrap.min.js"></script>

    </head>
    <body>
        <h1>Resultado de la comparación de los archivos</h1>

        <br>
             <%
            Controlador con=new Controlador();
            ArrayList entrada=new ArrayList();
            MiServlet ser=new MiServlet();
            String c= ser.doPost(request, response);
            String idioma="";
            double cosenob, cosenotf;
            float cos;


           String[] q={"Content-Disposition: ","form-data; ","name=\"
               file2\"; ",
              "filename=\"entrada4a.txt\"" ,"Content-Type:","text/plain
                  ","name=\"idioma\""};

           for (int i = 0; i < q.length; i++) {
            // guarda =" "+PalabrasVacias[i]+" ";
           c = c.replace("ES", "idEspañol");
           c = c.replace("IN", "idIngles");
           c = c.replace("text/plain", "pr1m3r0");
           c = c.replace(q[i], " ");
```

```
            }
              String[] noticias = c.split("pr1m3r0");

              String n1=noticias[0];

              String n2=noticias[1];
               if
                           (n2.contains("idEspañol")){
                           idioma="ES";
                         n2=n2.replace("idEspañol", "");
                         }

                         else{
                           idioma="IN";
                          n2=n2.replace("idIngles", "");
                         }

             entrada.add(n1);
             entrada.add(n2);

             //Se envian los textos a comparar
              cos=con.Ejecutar(entrada, idioma);

              //Recuperar los valores
               cosenob=con.getCosenob();
               cosenotf=con.getCosenotf();
               //pasar resultado a porcentajes
               cosenob*=100;
               cosenotf*=100;
               DecimalFormat df = new DecimalFormat("0.00");
               //Se muestra el resultado de la comparacion
               System.out.println("\nSimilitud␣por␣coseno␣binario:␣"+
                   cosenob);
               System.out.println("\n\nSimilitud␣por␣coseno␣TF:␣"+
                   cosenotf);

               out.println("\nSimilitud␣por␣coseno␣binario:␣"+df.format(
                   cosenob)+"%"+"</br>"+"</br>");
               out.println("\nSimilitud␣por␣coseno␣TF:␣"+df.format(
                   cosenotf)+"%"+"</br>"+"</br>");
               //out.println("\n\n"+n2);



        %>

        <br><br><br>
        <form  action="index.html"  method="post" >

                 <input type="submit" value="Ir␣a␣Inicio" /></form>


<div id="inst" class="container">
```

```
<!-- Trigger the modal with a button -->
<center>

<button type="button" class="btn␣btn-info␣btn-lg" data-toggle="modal"
    data-target="#myModal">Acerca de las métricas</button>
</center>
<!-- Modal -->
<div class="modal␣fade" id="myModal" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&
              times;</button>
          <h4 class="modal-title">Instrucciones de uso.</h4>
      </div>
      <div class="modal-body">
        <p><UL type = disk >

        </p>

        <ul>
            <li type="square">Coseno Binario
                <ol>
                    <li>Si la palabra del diccionario se encuentra en
                        el artículo se agrega un 1 en un vector
                        de enteros que representa verdadero, en caso
                            contrario se agrega un 0 para indicar
                        que no se encuentra la palabra.</li>

                </ol>
            </li>
            <br />
            <li type="square">Coseno TF
                <ol>
                    <li>Es una medida numérica que expresa cuán
                        relevante es una palabra para un documento en
                        una colección. </li>

                </ol>
            </li>
            <br />

        </ul>

      </div>
      <div class="modal-footer">
          <button type="button" class="btn␣btn-default" data-dismiss=
              "modal">Cerrar</button>
      </div>
    </div>

  </div>
</div>
```

```
</div>

    </body>
</html>
```

B.1.6. compararBD

```
<%--
    Document   : compararBD
    Created on : 2/03/2016, 03:46:17 PM
    Author     : Erik
--%>
<%@page import="conexionDB.ConexionDB"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>



<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>Comparar BD</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">


  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap
      /3.3.6/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
      jquery.min.js"></script>
  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
      bootstrap.min.js"></script>


    </head>
    <body>
        <h1>Resultados almacenados en la base de datos!</h1>

        <%
            String BD = request.getParameter("texto1");
            String idioma = request.getParameter("idioma");
            ConexionDB conexion= new ConexionDB();

            conexion.Conectar();
            conexion.leerDato(idioma);
            conexion.cerrarConexion();

        %>
        <br><br><br>
        <form  action="index.html"  method="post" >
```

```
                <input type="submit" value="Ir␣a␣Inicio" /></form>


<div id="inst" class="container">

  <!-- Trigger the modal with a button -->
  <center>

  <button type="button" class="btn␣btn-info␣btn-lg" data-toggle="modal"
      data-target="#myModal">Acerca de las métricas</button>
  </center>
  <!-- Modal -->
  <div class="modal␣fade" id="myModal" role="dialog">
    <div class="modal-dialog">

      <!-- Modal content-->
      <div class="modal-content">
        <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal">&
               times;</button>
            <h4 class="modal-title">Instrucciones de uso.</h4>
        </div>
        <div class="modal-body">
          <p><UL type = disk >

              </p>

             <ul>
              <li type="square">Coseno Binario
                  <ol>
                      <li>Si la palabra del diccionario se encuentra en
                           el artículo se agrega un 1 en un vector
                          de enteros que representa verdadero, en caso
                              contrario se agrega un 0 para indicar
                          que no se encuentra la palabra.</li>

                  </ol>
              </li>
              <br />
              <li type="square">Coseno TF
                  <ol>
                      <li>Es una medida numérica que expresa cuán
                          relevante es una palabra para un documento en
                           una colección. </li>

                  </ol>
              </li>
              <br />

          </ul>

        </div>
        <div class="modal-footer">
```

```
            <button type="button" class="btn btn-default" data-dismiss=
                "modal">Cerrar</button>
        </div>
      </div>

    </div>
  </div>

</div>

    </body>
</html>
```

B.1.7. compararTextos

```
<%--
    Document   : compararTextos
    Created on : 7/01/2016, 09:12:10 AM
    Author     : Erik
--%>


<%@page import="java.text.DecimalFormat"%>
<%@ page language="java" import="java.util.*,java.sql.*" pageEncoding=
    "ISO-8859-1"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.io.InputStreamReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="java.io.DataInputStream"%>
<%@page import="java.io.FileInputStream"%>


<%@page import="java.util.ArrayList"%>
<%@ page contentType="text/html; charset=utf-8"

         import="manipularArchivos.Controlador"

         errorPage=""%>




<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF
            -8">
        <title>Comparar textos</title>
        <link rel="stylesheet" type="text/css" href="Estilos.css">


  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap
    /3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
    jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
    bootstrap.min.js"></script>

  </head>
  <body>
      <h1>El resultado de la comparación es: </h1>

      <br>
      <%
     try
     {
             Controlador con=new Controlador();
             ArrayList entrada = new ArrayList();
             String art1 = request.getParameter("texto1");
             String art2 = request.getParameter("texto2");
             double cosenob, cosenotf;
             float cos;
             entrada.add(art1);
             entrada.add(art2);
             String idioma = request.getParameter("idioma");

            //Se envian los textos a comparar
             cos=con.Ejecutar(entrada, idioma);

             //Recuperar los valores
             cosenob=con.getCosenob();
             cosenotf=con.getCosenotf();
             //pasar resultado a porcentajes
             cosenob*=100;
             cosenotf*=100;
             DecimalFormat df = new DecimalFormat("0.00");
             //Se muestra el resultado de la comparacion


             System.out.println("\nSimilitud␣por␣coseno␣binario:␣"+
                 cosenob);
             System.out.println("\n\nSimilitud␣por␣coseno␣TF:␣"+
                 cosenotf);

             out.println("\nSimilitud␣por␣coseno␣binario:␣"+df.format(
                 cosenob)+"%"+"</br>"+"</br>");
             out.println("\nSimilitud␣por␣coseno␣TF:␣"+df.format(
                 cosenotf)+"%"+"</br>"+"</br>");
      }
       catch(Exception e)
     {
               out.println(e.getMessage());
       }
       %>

        <form action="index.html" method="post" >
```

```
                <input type="submit" value="Ir a Inicio" /></form>


<div id="inst" class="container">

  <!-- Trigger the modal with a button -->
  <center>

  <button type="button" class="btn btn-info btn-lg" data-toggle="modal"
      data-target="#myModal">Acerca de las métricas</button>
  </center>
  <!-- Modal -->
  <div class="modal fade" id="myModal" role="dialog">
    <div class="modal-dialog">

      <!-- Modal content-->
      <div class="modal-content">
        <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal">&
                times;</button>
            <h4 class="modal-title">Instrucciones de uso.</h4>
        </div>
        <div class="modal-body">
          <p><UL type = disk >

              </p>

              <ul>
              <li type="square">Coseno Binario
                  <ol>
                      <li>Si la palabra del diccionario se encuentra en
                          el artículo se agrega un 1 en un vector
                          de enteros que representa verdadero, en caso
                              contrario se agrega un 0 para indicar
                          que no se encuentra la palabra.</li>

                  </ol>
              </li>
              <br />
              <li type="square">Coseno TF
                  <ol>
                      <li>Es una medida numérica que expresa cuán
                          relevante es una palabra para un documento en
                          una colección. </li>

                  </ol>
              </li>
              <br />

          </ul>

        </div>
        <div class="modal-footer">
```

```
            <button type="button" class="btn␣btn-default" data-dismiss=
                "modal">Cerrar</button>
        </div>
      </div>

    </div>
  </div>

</div>

    </body>
</html>
```

B.1.8. Estilos

```
/*
    Created on : 7/03/2016, 05:07:29 PM
    Author     : Erik
*/

 h2{
    text-align: center;
   }
 h1,h3 {
text-align: center;
color: #323133;
font-family: Helvetica Neue, Arial, Helvetica, sans-serif;
letter-spacing: -1px;
text-decoration: none;
text-shadow: 1px 1px #fff, 0 0 #0e0e0e, 2px 3px 1px #e3e3e3;
text-transform: none;
word-spacing: -2px;
}
body{

  background-image: url("imagenes/4.jpg");

}
.container{

    left:50%;
    top:50%;
}

CSS

/*Eliminamos los margenes y paddings que agrega el navegador por
    defecto*/
* {
  padding: 0;
  margin: 0;
}
```

```
/*Agregamos margenes inferiores a los parrafos*/
p {
  margin-bottom: 20px;
}

nav {
  float: left; /* Desplazamos el nav hacia la izquierda */
}

nav ul {
  list-style: none;
  overflow: hidden; /* Limpiamos errores de float */
}

nav ul li {
  float: left;
  font-family: Arial, sans-serif;
  font-size: 16px;
}

nav ul li a {
  display: block; /* Convertimos los elementos a en elementos bloque
     para manipular el padding */
  padding: 20px;
  color: #fff;
  text-decoration: none;
}

nav ul li:hover {
  background: #3ead47;
}

.contenido {
  padding-top: 80px;
}

.wrapper {
  width: 80%;
  margin: auto;
  overflow:hidden;
}

.container {
  padding-top: 80px;
}
```

Bibliografía

[1] MORA R. *Sistema de detección de plagio en archivos de texto*, Universidad Autónoma Metropolitana. (2013).

[2] ALEJANDRO F. *Sistema de procesamiento de textos de investigación*, Universidad Autónoma Metropolitana.(2013)

[3] SANTILLÁN S. *Sistema de agrupamiento de servicios web semánticos utilizando un algoritmo bioinspirado*, Universidad Autónoma Metropolitana.(2014)

[4] KIRK D. *Identifying Similarity in Text: Multi-Lingual Analysis for suummarization*, Columbia University (2005).

[5] DANUSHKA BOLLEGALA, YUTAKA MATSUO and MITSURU ISHIZUKA *A Web Eearch Engine-Bassed Approach to Measure Semantic Similarity Between Words*, TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 23, no. 7, July 2011. IEEE.

[6] *SEMILAR: A Semantic Similarity Toolkit.*, Disponible en: http ://www.semanticsimilarity.org/ . Última revisión 8 de diciembre del 2014.

[7] *Búsqueda Y Recuperación De Información | E-Coms. (n.d.).*, Revisado el 07 de marzo, 2016 de http://www.mariapinto.es/e-coms/busqueda-y-recuperacion-de-informacion/

[8] ANDREAS SCHMIDT & NIKLAS SKAMRIIS *Textual Similarity: Comparing texts in order to discover how closely they discuss the same topics*, Technical University of Denmark (2008).

[9] BAZARAA, M.S., J.J. JARVIS y H.D. SHERALI, *Programación lineal y flujo en redes*, segunda edición, Limusa, México, DF, 2004.

[10] DANTZIG, G.B. y P. WOLFE, «Decomposition principle for linear programs», *Operations Research*, **8**, págs. 101–111, 1960.