UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TESIS

DESARROLLO DE UN EXTRACTOR DE PALABRAS CLAVE

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

VÍCTOR HUGO SÁNCHEZ CASTILLO

MARCO ANTONIO VELÁZQUEZ HERRERA

DIRECTOR DE TESIS:

DR. ALFONSO MEDINA URREA

CIUDAD UNIVERSITARIA AGOSTO 2013.

Agradecimientos

Esta tesis se llevó a cabo mediante becas del proyecto 105711 del CONACyT (*Extracción de conocimiento lexicográfico a partir de textos de Internet*, Convocatoria Ciencia Básica 2008), bajo la responsabilidad del Dr. Alfonso Medina Urrea (hasta julio de 2012) y del Dr. Servio Tulio Guillén Burguete (después de julio 2012).

Un agradecimiento especial al director de este proyecto el Dr. Alfonso Medina Urrea, por guiarnos durante el proceso de diseño e implementación de la aplicación, además de sus acertadas sugerencias y correcciones en la redacción de esta tesis. También agradecemos a los sinodales por invertir parte de su tiempo en la revisión de este trabajo y colaborar con su opinión para mejorarlo.

Por último damos las gracias a nuestros familiares y amigos por su apoyo, no solo a lo largo del desarrollo de esta tesis, sino también, durante el transcurso de nuestra carrera. Sin ellos, nada de esto hubiera sido posible.

Índice de contenido

Índice de tablas

Índice de figuras

Capítulo 1. Introducción

Definición del problema

En la actualidad, experimentamos una explosión tecnológica y científica en todas las áreas de la actividad humana, la existencia de herramientas de difusión y colaboración como el internet ha facilitado la generación de nuevo conocimiento y ha permitido un acceso más rápido a dicho conocimiento.

La capacidad de nuestra sociedad de compartir y colaborar en proyectos de otros lugares del mundo ha tenido un efecto importante en las comunidades científicas y de investigación que generan enormes cantidades de información a través de artículos, tesis, libros y otros documentos. Los artículos son especialmente útiles para determinar las tendencias de determinado campo del conocimiento debido a su brevedad relativa y fácil acceso.

Desafortunadamente, algunos campos producen demasiados artículos de muchas fuentes distintas, lo que dificulta la labor de analizar y clasificar tales documentos. El uso de palabras clave se volvió una práctica usual en la publicación de artículos de divulgación, como un elemento para identificar el contenido principal y general de la publicación, así como elementos vitales en la búsqueda y clasificación en grandes grupos de documentos.

Surge ahora la necesidad de determinar el grupo de palabras clave que represente de mejor forma el contenido del artículo; una elección errónea o descuidada de dichas palabras podría resultar en una difusión pobre del contenido del documento. Dicha necesidad viene con la problemática de determinar las características y propiedades que convierten a un término dentro de un documento en una palabra clave; es decir, se requiere diseñar una técnica eficaz en la extracción de términos relevantes, candidatos a palabras clave, de cualquier texto o publicación electrónica.

En la actualidad existe el proyecto: “Extracción de conocimiento lexicográfico a partir de textos de Internet” coordinado por el Consejo Nacional de Ciencia y Tecnología (CONACyT), (Convocatoria Ciencia Básica 2008, 105711), que entre otras funciones busca la extracción terminológica a partir de un corpus de documentos sobre sexualidad en México.

Además, existe el proyecto LINO-TAR, dirigido por Eugenio Mario López Ortega, investigador del Instituto de Ingeniería, que se enfoca en herramientas de inteligencia tecnológica para determinar las tendencias de un área, específicamente el tratamiento de aguas residuales. El proyecto consiste en seguir los artículos de varias publicaciones y almacenar las palabras clave; dicho proceso, generalmente, se ejecuta de forma manual y confiando ciegamente en la veracidad de las palabras claves definidas por los autores, además de que no siempre son incluidas con la publicación del archivo.

Objetivos

El objetivo principal de esta tesis es utilizar los métodos de minería de textos para el desarrollo de una aplicación, que sea capaz de identificar las palabras clave (PC) de un conjunto de documentos. Los métodos utilizados determinarán la importancia de secuencias de palabras (que llamaremos expresiones multipalabra) respecto a las demás de cada documento. Otro objetivo importante es establecer los parámetros que definan a las palabras clave con el objeto de obtener un método eficaz de extracción de PC de cualquier documento. Cabe señalar que nuestra aplicación no utilizará ningún tipo de etiquetador gramatical, ya que buscamos resolver el problema por medio de métodos enteramente estadísticos.

La aplicación recibirá un conjunto de documentos como entrada y el resultado de procesarlos será una lista de términos, considerados como los más representativos de la información analizada. Esta aplicación tendrá dos modos de funcionamiento:

1. El primero consiste en la generación de las PC de un documento perteneciente a una línea de investigación específica. En este modo, se utiliza conjuntamente una lista de PC del tema que trata el documento, la cual ya ha sido verificada por especialistas, es decir, la aplicación buscará las PC apoyándose de esta lista.

2. Finalmente se busca desarrollar y poner a la disposición del público en general, un extractor de PC genérico que funcione sobre cualquier documento.

Lo interesante es que el resultado del programa propuesto puede ser utilizado para muchos fines, como por ejemplo la clasificación y agrupamiento de los documentos, para ver las tendencias que tienen las líneas de investigación sobre cierto tema en específico.

Estado del arte

La extracción automática de palabras claves es un campo en constante desarrollo que ha tomado gran importancia en distintas áreas del conocimiento humano.

En la actualidad, existen grandes colecciones de documentos, artículos y recursos textuales que sirven distintos propósitos, para mantener un control riguroso de tales colecciones es necesario emplear herramientas automáticas de análisis y procesamiento de textos. Por tal motivo, en todo el mundo, se realizan investigaciones orientadas a crear y mejorar las técnicas y metodologías usadas en la extracción de palabras claves.

El problema de la extracción de palabras claves ha generado una gran cantidad de aproximaciones para resolverlo, siendo las más comunes el uso de métodos estadísticos, de aprendizaje y simbólico-estadísticos (híbridos).

Los métodos estadísticos se caracterizan por ser sencillos de implementar y ofrecer resultados aceptables; la medida TF-IDF (*term frequency-inverse document frequency*) usada de forma exitosa para extraer palabras clave de un documento, teniendo como contexto un conjunto de documentos relacionados, lo que lo limita, ya que necesita una colección confiable para funcionar.

El trabajo de Matsuo e Ishizuka (2004) utiliza información estadística de los términos obtenida a través de matrices de co-ocurrencia de palabras del documento, determinando la tendencia a partir del uso de χ^2, lo que presenta grandes ventajas sobre TF-IDF ya que no requiere el uso de colecciones de documentos. Toda la información se extrae del documento mismo.

Además del trabajo de Matsuo e Ishizuka se ha experimentado con el uso de otros métodos estadísticos para extraer palabras clave, un ejemplo es el trabajo de Sidorov *et al*. (2010) que, utilizando un corpus de referencia general, compara el documento procesado con el corpus de determinado tema y, mediante el cálculo de la razón de semejanza (*log-likelihood*), obtiene un índice que le permite,

después clasificar los posibles términos con técnicas de reconocimiento de patrones para obtener una lista de palabras clave.

El método del C-value/NC-value (Frantzi *et al*. 2000), por su parte, se considera un método híbrido (simbólico-estadístico) porque realiza un procesamiento de tipo lingüístico al documento antes de un análisis estadístico, siendo parte de la etapa lingüística el uso de etiquetado POS y el uso de listas de paro; para después proceder con el cálculo del C-value que establece una medida de la cualidad de una palabra o grupos de palabras de ser un término en algún lenguaje de especialidad (Ananiadou y McNaught,2006, p.76), cualidad que podemos llamar "terminidad" (*termhood*) de las palabras del documento.

A partir del C-value surgió el MC-value o Modified C-value (Nakagawa y Mori, 2002) que es capaz de calcular la "terminidad" de términos de una sola palabra, lo que es una mejora con el C-value que, debido a su definición matemática no era capaz de evaluar la "terminidad" de términos sencillos.

Respecto al uso de aprendizaje supervisado en la extracción de palabras clave, se puede crear un sistema de aprendizaje al que se entrene para reconocer palabras clave en un documento. Tal aproximación fue propuesta originalmente por Peter D. Turney en 1999 a través del algoritmo GENex (Turney, 1999), que combinaba el Algoritmo Genético de Estado Estable (Whitley, 1989) con el Algoritmo de Extracción Parametrizada de Frases Clave (*parameterized keyphrase extraction algorithm*) del mismo Turney, el cual ofrecía resultados con 60% de efectividad en frases clave evaluadas como buenas y hasta 80% con frases aceptables.

Después del trabajo de Turney se generaron varias propuestas para la extracción de palabras clave utilizando distintas aproximaciones en el campo del aprendizaje supervisado y utilizando parámetros numéricos para la evaluación de las posibles palabras clave.

Otro avance se dio en el 2003, cuando Anette Hulth de la Universidad de Estocolmo, combinó elementos tanto léxicos como sintácticos para mejorar los

algoritmos de aprendizaje supervisado para la extracción de palabras clave, todo esto a través del uso de etiquetado de categorías gramaticales POS (*Parts-Of-Speech*) para evaluar patrones empíricos en términos multipalabra; esta nueva aproximación mejoró la efectividad de la extracción de palabras clave más allá de cualquier otro algoritmo publicado con anterioridad (Hulth, 2003).

A partir del trabajo de Hulth, surgió el Algoritmo de Clasificación de Texto basado en Grafos (Mihalcea y Tarau, 2004). Este nuevo algoritmo utiliza un grafo no supervisado de palabras y sus distancias para asignar un peso a las distintas unidades léxicas encontradas en el preprocesamiento del documento, lo que mejora los resultados obtenidos por Hulth, que utiliza aprendizaje supervisado.

El uso de grafos no supervisados para la extracción de palabras clave ha generado una serie de proyectos paralelos donde se experimenta evaluando distintas características en los vértices del grafo, como lo demuestra el trabajo de Inoue y McCracken (2010), donde utilizan una aproximación diferente al algoritmo de Mihalcea y Tarau que modifica los valores del grafo utilizando un índice de relaciones semánticas como arcos o transiciones del grafo en lugar de representar la simple co-ocurrencia, planteada originalmente, con lo que se logran buenos resultados para el área de recursos educativos.

Método

La minería de textos cuenta con un gran número de herramientas para la extracción automática de información, cada herramienta trabaja con texto en distintos niveles de complejidad, de manera general se pueden enumerar los siguientes:

- Léxico: Trata a las palabras y términos como elementos individuales independientes de su contexto, enfocándose en sus propiedades numéricas y estadísticas.
- Sintáctico: Agrupa palabras y términos en oraciones o fragmentos de oraciones, donde cada desempeña una función dentro de la misma oración.
- Semántico: Intenta determinar el significado de las oraciones o de un conjunto de oraciones a través de un análisis profundo del texto.

En el desarrollo de proyectos similares, generalmente se utilizan herramientas de análisis léxico de los términos en el documento. Algunas veces se utilizan, de forma limitada, herramientas para un análisis sintáctico parcial, sólo para apoyar o reforzar los resultados del análisis léxico.

Nuestra aplicación realizará un análisis del texto sólo a un nivel léxico ya que sus funciones principales consisten en el tratamiento de términos individuales; aún cuando se trate de términos multipalabra, el tratamiento no es diferente a los términos sencillos de una palabra. Consideramos que es necesario profundizar en este nivel antes de investigar los otros niveles de análisis para conseguir el objetivo de la aplicación. Tal objetivo se puede llevar a cabo realizando las siguientes acciones:

1. Pre-procesamiento de los documentos.

En esta etapa la aplicación guarda el documento original y lo separa en enunciados (lo que hay entre punto y punto). Utiliza los primeros enunciados del documento para determinar el idioma en el que está escrito (actualmente sólo identifica los

idiomas español e inglés; este último por ser el más utilizado en documentos de carácter científico). Después separa cada enunciado en palabras gráficas, es decir unigramas, y se guardan luego de haber sido pasadas por un proceso de lematización. Para esto, utilizamos el *Porter Stemmer Algorithm.* Análogamente guardamos los bigramas (dos unigramas); trigramas (tres unigramas); cuatrigramas o tetragramas (dos bigramas) y pentagramas (cinco unigramas).

2. Cálculo de la relevancia de términos.

Una vez que tenemos guardado el documento en n-gramas lematizados, se procede a calcular la relevancia de cada uno mediante los siguientes valores. En el siguiente capítulo se ofrece un análisis sobre la obtención de cada uno.

- Medida AM (*Association Measure*) y razón de semejanza (*Likelihood Ratio*): ofrecen la relación que existe entre palabras, considerando la correlación que existe entre ellas.

- TF-IDF (*Term Frequency Inverse Document Frequency*) y TF-IPF (*Term Frequency Inverse Paragraph Frequency*): indican la relevancia de una palabra dependiendo de su frecuencia y los lugares en los que aparece. La primera medida lo hace respecto a un conjunto de documentos pertenecientes a un mismo tema, mientras que la segunda lo hace respecto a un conjunto de párrafos de un documento.

- Índice de gramaticalidad (IG): esta lista nos muestra n-gramas que no contienen palabras funcionales.

- C-value (Cv): parte estadística del C-value/CN-value que favorece la relevancia de multipalabras.

3. Presentación de resultados.

Como resultado, el sistema mostrará una lista de palabras clave que se consideran representativas del documento.

Mapa de la tesis

El presente capítulo (1) analiza el problema de la extracción de palabras clave en el contexto académico. Muestra diversas aproximaciones utilizadas para la extracción de información mediante diferentes métodos y aplicaciones utilizadas en la actualidad. Teniendo un punto de vista general sobre el problema, se ofrece una explicación sobre el método propuesto para el desarrollo de la aplicación de extracción de PC.

Una introducción a la minería de datos, haciendo énfasis especial en la minería de textos, se presenta en el capítulo 2. Éste también cuenta con la definición de la terminología y explicación de fórmulas y algoritmos empleados en el presente trabajo, mientras que la información correspondiente al software utilizado para desarrollar la aplicación, se detalla en el capítulo 3.

La parte medular de la tesis se especifica en el capítulo 4. Es en este capítulo donde se describen detalladamente los pasos para la extracción de palabras clave, es decir, cómo se procesa el documento para obtener los candidatos a palabras clave, qué parámetros se toman en cuenta para medir la relevancia de los candidatos, para finalmente presentar los resultados.

El extractor de palabras clave desarrollado se probó con información de los proyectos mencionados anteriormente. Del proyecto de “Extracción de conocimiento lexicográfico a partir de textos de Internet” se tomó un artículo de sexualidad en español para su análisis, mientras que del proyecto LINO-TAR se tomó una colección de resúmenes (*abstracts*) en inglés como un solo documento para su posterior análisis. Los resultados obtenidos, son analizados en el capítulo 6, donde se discuten las posibles combinaciones de las medidas para determinar la relevancia de los candidatos a término de nuestra aplicación, dando paso a la elección de una. Además se realiza una comparación entre nuestros resultados respecto a los de otras aplicaciones.

Finalmente, en el capítulo 7 se hace mención de los resultados obtenidos y las posibles mejoras.

Capítulo 2. Marco teórico

Minería de datos

El término “minería de datos” es el nombre dado al conjunto de técnicas y herramientas diseñadas para la extracción automática de conocimiento de un conjunto de datos estructurados explícitamente, sobre todo en forma tabular. La extracción de conocimiento a partir de un gran conjunto de datos ha generado distintas opiniones sobre si el término “minería de datos” representa las distintas facetas de la disciplina:

> Para referirnos a la extracción de oro de las rocas o la arena, decimos minería de oro no de roca o arena. De forma análoga, la minería de datos, debería de ser nombrada, de forma más apropiada ‘Minería de conocimiento a partir de datos’ el cual es desafortunadamente largo (Han, Kamber, Pei, 2011, p. 5; traducción nuestra).

Así se puede entender que el objetivo de la minería de datos es, básicamente, la extracción de información útil (conocimiento) a partir de un conjunto muy grande de datos, de la misma manera que los minerales son extraídos de entre una gran cantidad de material no deseado como rocas y arena.

Debido a la ambigüedad de su definición, varios autores han definido distintos nuevos términos para nombrar a estas herramientas, entre los que destacan: *extracción de conocimiento, análisis patrón/datos, arqueología de datos* o *dragado de datos.*

El término “descubrimiento de conocimiento a partir de datos”, o KDD por sus siglas en inglés (*Knowledge Discovery from Data*), es comúnmente usado como sinónimo para la minería de datos, aunque hay quienes definen a la minería de datos como una fase del proceso de KDD.

De forma general, la tecnología de la minería de datos puede ser usada para cualquier tipo de datos, siempre y cuando, estos datos tengan un significado

para la aplicación objetivo. Los ejemplos más comunes del uso de minería de datos son su uso en las bases de datos, los almacenes de datos y en los datos de transacciones, aunque estas técnicas también pueden aplicarse a otros tipos de datos como flujos de datos, datos espaciales y datos de texto.

En los últimos años, se han dado grandes avances en el uso de minería de datos de textos o minería de textos, ya que nos permite obtener conocimiento a partir de enormes conjuntos de datos textuales, como libros, artículos, blogs y comentarios. Como lo especifican Han, Kamber y Pei (2011):

> Al usar la minería en datos de texto, como la literatura sobre minería de datos de los pasados diez años, podemos identificar la evolución de temas importantes en el campo. Al analizar comentarios de usuarios sobre productos (que usualmente se hacen en forma de mensajes de texto cortos), podemos valorar las opiniones del cliente y comprender qué tan bien un producto es aceptado por un mercado (p. 14; traducción nuestra).

De esta forma, las herramientas de minería de datos se hacen disponibles para todo tipo de datos, donde las técnicas de la minería se adaptan a las distintas configuraciones de datos. En el caso particular de los datos de texto, un campo completo de investigación se ha desarrollado con el fin de poder elaborar técnicas de minería de datos que se puedan aplicar eficazmente al análisis de datos no estructurados en forma tabular, como lo es el lenguaje escrito.

Minería de textos

La minería de textos busca descubrir y extraer conocimiento útil a partir de información estructurada en forma tabular y aquella contenida en textos, aplicando métodos automáticos para analizarla y estructurar los datos que contiene. Principalmente se enfoca en el descubrimiento de patrones interesantes y conocimiento nuevo que puede haber en un conjunto de textos. Su objetivo es descubrir nuevas tendencias, desviaciones y asociaciones dentro de grandes volúmenes de infor-

mación textual (Kao *et al* 2007). A continuación presentaremos varios conceptos centrales de la minería de textos que se presentan en Weiss *et al*. (2005).

La minería de textos cuenta con un gran número de herramientas para la extracción automática de información, cada herramienta trabaja con el texto en distintos niveles de complejidad. De manera general se pueden enumerar los siguientes:

- Léxico: Trata a las palabras y términos como elementos individuales independientes de su contexto, enfocándose en sus propiedades numéricas y estadísticas.

- Sintáctico: Agrupa palabras y términos en oraciones o fragmentos de oraciones, donde cada uno desempeña una función dentro de la misma oración.

- Semántico: Intenta determinar el significado de las oraciones o de un conjunto de oraciones a través de un análisis profundo del texto.

Se presentan a continuación los principales usos de la minería de textos:

Clasificación de documentos

Consiste en asignar un documento a una o más clases o categorías previamente definidas; es decir, cuando se presenta un documento éste se analiza y se determina de qué tema o temas trata.

La clasificación automática de documentos ha obtenido mayor interés a medida que se extiende el uso del Internet. En los últimos años se han desarrollado diversas técnicas utilizando: redes neuronales, algoritmos genéticos, correlación difusa, árboles de decisión, redes bayesianas, k-vecinos más cercanos (KNN), entre otros. (Khan *et al* 2009).

Actualmente dentro de sus aplicaciones más importantes se encuentra la identificación del tema e idioma de un texto; así como la detección de correos electrónicos no deseados.

Recuperación de información (RI)

Un sistema RI guarda y maneja información contenida en documentos, generalmente documentos en línea. El sistema ayuda a los usuarios a encontrar la información que necesitan, al indicar la existencia y ubicación de documentos que pueden contener la información buscada.

El usuario ingresa al sistema, siendo éste habitualmente un motor de búsqueda, palabras que identifican el tema de su interés. Algunos documentos sugeridos satisfarán la búsqueda del usuario, por lo que se denominan *documentos relevantes*. Un sistema RI perfecto, sólo mostraría documentos relevantes sin mostrar aquellos que no lo son, sin embargo estos sistemas no existen ya que en la mayoría de las ocasiones la relevancia de la información depende de una opinión subjetiva del usuario.

Un concepto básico para la recuperación de información es medir la similitud entre documentos. Generalmente se realiza entre dos de ellos, pero también se pude considerar, para fines de búsqueda, al conjunto de palabras ingresadas por el usuario como un documento. Las principales medidas de similitud son: comparación semántica (cantidad de palabras que comparten, palabras de diccionario que comparten) y comparaciones matemáticas (coeficiente cosenoidal) (Göker *et al*, 2009, p.6)

Agrupación y organización de documentos

Para la clasificación de documentos, el objetivo es asignarles el tema que les corresponde, estos temas son establecidos antes de realizar el proceso por alguien que conoce de cierta forma el contenido de los documentos.

La agrupación de documentos funciona de la siguiente forma: a partir de un conjunto de documentos busca encontrar características que algunos de ellos

compartan, para poder agruparlos u organizarlos de acuerdo a éstas. La agrupación es similar a la asignación de los temas necesarios para la clasificación de documentos, es decir, la aplicación encuentra los temas que tratan los documentos sin necesidad de establecerlos manualmente.

Extracción de información (EI)

Se puede representar un sistema de Extracción de información de dos maneras: 1) tomando un texto en lenguaje natural y extraer los hechos esenciales sobre un tema predefinido; ó 2) tomando la representación de cada hecho como una tabla cuyas celdas son llenadas con base en la información que se encuentra en el texto.

Estos sistemas extraen información acerca de un dominio específico de un texto escrito en lenguaje natural. El tema y tipo de información a ser extraída debe ser definida antes de llevar a cabo el proceso, es decir, busca extraer información que está semánticamente definida en el texto y guardarla en una base de datos previamente dada (Moens, 2006, p.12).

Para facilitar el análisis de los textos, se les puede dar un tratamiento previo, el cual puede consistir en, cuando menos: segmentación del texto y lematización. Estos conceptos serán revisados en la siguiente sección.

Segmentación del texto

El primer paso para el procesamiento de texto es dividirlo en sus partes más elementales, ya que esto facilitará su estudio posterior. Este proceso que es denominado en inglés *tokenization,* consiste en dividir el texto en unidades, que corresponden a palabras gráficas en el idioma en el que está escrito el texto.

Al momento de llevar a cabo la segmentación del texto de forma automática, se debe poner especial atención en algunos signos de puntuación, como el punto, debido a que puede indicar el final de una oración, la precisión de un número o el final de una abreviatura.

N-gramas

Un n-grama es una serie de n elementos de una secuencia dada. Son utilizados ampliamente en el procesamiento estadístico del lenguaje natural. El término n-grama fue introducido por Claude Shannon en 1948 (Shannon, 1948). Los n-gramas que contienen 1, 2, 3, 4, o *n* elementos son llamados unigrama, bigrama, trigrama, tetragrama y n-grama respectivamente. En nuestro caso consideramos a los n-gramas como conjunto de palabras gráficas del texto.

Lematización

Una parte del procesamiento de texto es la lematización. Esta técnica usualmente es aplicada después de haber separado las cadenas de texto en conjuntos de palabras gráficas relacionadas entre sí (lápiz, lápices), ya que busca representar cada conjunto en su forma canónica (lápiz). El efecto que tiene esto es reducir el número de palabras distintas en un texto, lo que conlleva al aumento de la frecuencia de su forma canónica.

Lingüísticamente, la lematización consiste en hallar el lema o forma de diccionario de una palabra flexionada; es decir, de una forma que se encuentra en plural, en masculino, en femenino o es un verbo conjugado. Por ejemplo, las pala-

bras *satisfaces, satisfizo, satisfaga, satisficiesen* se reducirían al lema *satisfacer*. Normalmente, este proceso se realiza con ayuda de diccionarios.

La importancia de este paso depende del uso que se le quiera dar a la información. Para la clasificación de documentos puede resultar en un aumento positivo del resultado de la clasificación, en especial si se utilizan algoritmos que toman en cuenta la frecuencia.

Dado que en la mayoría de las ocasiones no se requiere un proceso de lematización que sea lingüísticamente correcto, surgieron métodos que eliminan las flexiones gramaticales de una palabra sin tomar en cuenta su significado. Por lo tanto, palabras como *verde* y *verdad* podrían, según el algoritmo, reducirse a *verde*.

Este tipo de procesos son muy útiles en aplicaciones de recuperación de información. El método quizá más conocido es el algoritmo de Porter, que será utilizado en este trabajo y se describirá a continuación.

El algoritmo de Porter

Fue escrito en 1979 por Martin Porter en los laboratorios de computación de la Universidad de Oxford en el Reino Unido (Porter, 1979). Es un proceso para la eliminación de las terminaciones gramaticales de palabras escritas en inglés. Su finalidad es dejar únicamente la base sin los sufijos de una palabra.

Por ejemplo, las palabras *connected, connecting, connection* y *connections*, después de ser procesadas por el algoritmo, son reducidas al morfema *connect*. Esto se hace mediante la eliminación de los sufijos *-ed, -ing, -ion, ions* para dejar el término *connect*.

Procesamiento del texto

Para hacer el procesamiento del texto, nos basamos en la tabla de contingencia de bigramas presentada por Kageura (1999). Además utilizamos 5 medidas estadísticas para poder determinar la relevancia de los n-gramas obtenidos en la segmentación del texto. Tanto la tabla de contingencia como las medidas empleadas se describirán a continuación.

Tabla de contingencia

Una tabla de contingencia es una representación de datos en una clasificación de doble entrada, los datos se clasifican en celdas y se reporta cuántos hay en cada una de ellas. La tabla de contingencia sirve para determinar si los datos muestran que las dos variables son dependientes o independientes.

Se muestra a continuación una tabla de contingencia para bigramas formados por la palabra uno w_1 y la palabra dos w_2.

	w_2	$\overline{w}_2$	*Total*
w_1	$f_{11} = f(w_1 w_2)$	$f_{12} = f(w_1 \overline{w}_2)$	$f_{1.} = f(w_1)$
$\overline{w}_1$	$f_{21} = f(\overline{w}_1 w_2)$	$f_{22} = f(\overline{w}_1 \overline{w}_2)$	$f_{2.} = f(\overline{w}_1)$
Total	$f_{.1} = f(w_2)$	$f_{.2} = f(\overline{w}_2)$	$f_{..} = \sum f(w_i)$

Tabla 1. Tabla de contingencia de colocación de bigramas (Kageura 1999).

Se tiene entonces que los renglones corresponden a la presencia de la primera palabra del bigrama, siendo w_1 su presencia y, $\overline{w}_1$ su ausencia. A su vez las

columnas corresponden a la segunda palabra w_2 cuando está presente y $\overline{w}_2$ cuando no.

Además, las funciones representan la frecuencia en que ocurren:

- f_{11}: ambas,
- f_{12}: sólo la primera,
- f_{21}: sólo la segunda y
- f_{22}: ninguna.

También se tiene la frecuencia de cada palabra:

- $f_{1.}$: de la primera,
- $f_{.1}$: complemento de la primera,
- $f_{2.}$: de la segunda y
- $f_{.2}$: complemento de la segunda

A menudo los términos que caracterizan a un documento son de una palabra. Sin embargo, existen muchísimos casos en los que es de gran ayuda considerar a un grupo de palabras como tal. Un ejemplo es el caso de nombres de entidades como *Universidad Nacional Autónoma de México*.

Para medir la relación que existe entre palabras, se suele considerar la correlación que existe entre ellas.

Medidas empleadas

En este proyecto para medir la asociación entre las palabras de un bigrama, utilizamos la razón de semejanza (*log likelihood*), la asociación AM (Ananiadou y McNaught, 2006, p.34), la medida C-value, un índice de gramaticalidad y una variante de Tf-Idf aplicada a párrafos en lugar de documentos. Estas medidas se describen a continuación.

Razón de semejanza

Esta medida estadística es ampliamente conocida en procesamiento del lenguaje natural; véase Jurafsky y Martin (2009) para mayor información. En el trabajo presentado por Kageura (1999) se examinan cuatro medidas estadísticas para el tratamiento de información: χ^2 *(ji cuadrada), coeficiente de coligación de Yule, información mutua* y *razón de semejanza*.

La razón de semejanza tuvo un mejor desempeño en comparación con las demás medidas, al momento de existir una variación de la frecuencia de las palabras. De este modo, para la tabla de contingencia de bigramas se calcula la razón de semejanza de la siguiente forma:

$$-2\,log\,\lambda\; =\; 2\,[\Sigma\,log\,L\,(f1c\,/\,f.c\,,f1c,f.c) - \Sigma\,log\,L\,(f1./\,f..,f1c,f.c)$$

donde: $logL(p,n,k)\; = k\,log(p)\; +\; (n-k)log(1-p)$

Razón de semejanza (Dunning 1993)

Una explicación detallada sobre la razón de semejanza para bigramas es tratada por Dunning (1993).

Medida de asociación AM

Como su nombre lo indica, esta medida muestra qué tan asociado estadísticamente está un bigrama:

$$AM = \frac{size(T)\log_{10}(f(T))f(T)}{\sum_{palabra_{i\epsilon T}} f(palabra_i)}$$

Medida de asociación AM (Weiss *et al* 2005:34)

Donde AM es la medida de asociación de la multipalabra T , $size(T)$ representa el número de palabras que contiene y $f(T)$ la cantidad de veces que T ocurre en la colección de documentos.

Identificación de términos usando TF-IDF

TF (*term frequency*) es la frecuencia de una palabra en un texto. Si la palabra tiene un TF alto en varios textos, es difícil conocer qué características del texto son representadas por esta palabra. Por lo tanto, usar sólo este valor para identificar términos presenta muchas limitaciones. Se representa la frecuencia de la palabra j en un documento mediante: $TF(j)$

En 1972, Spark Jones propuso que el calcular la cantidad de documentos en los que ocurre una palabra *(DF Document Frequency)*, sería de gran ayuda para asignarles un peso. Desde entonces el uso de la frecuencia inversa del documento *(IDF Inverse Document Frequency)* ha jugado un papel importante en la extracción de información, ya que reduce la importancia de términos que aparecen en la mayoría de los documentos, mientras que aumenta la de aquellos que están en un número menor. El IDF de la palabra j se calcula de la siguiente manera:

$$IDF(j) = \log\left(\frac{N}{DF(j)}\right)$$

Donde N es la cantidad de documentos analizados y DF es el número de documentos en los que aparece la palabra j.

Finalmente ambos se multiplican para formar el TF-IDF. Así, el TF-IDF de la palabra *j* es:

$$TF - IDF\ (j) = TF(j) * IDF(j)$$

Existen versiones alternativas a esta fórmula, pero el principio de funcionamiento es el mismo, el cual es obtener un valor para determinar qué tan bien caracteriza al documento cierta palabra, es decir, qué tan relevante es. Debido a que en algunas ocasiones sólo se cuenta con un documento, se hizo la siguiente modificación a la fórmula original:

$$TF - IPF\ (j) = TF(j) * IPF(j)$$

Donde $TF(j)$ representa el número de ocurrencias del n-grama j en el documento, mientras que $IPF(j)$ muestra la frecuencia inversa del número de párrafos que contienen el n-grama j. Con esta modificación se puede determinar la relevancia de un n-grama en un documento.

C-value

El *C-value* es una medida estadística que determina un valor de "terminidad" (del que hablamos en la introducción de esta tesis) de una expresión multipalabra candidata a PC. La medida se calcula usando características estadísticas de la cadena multipalabra (Frantzi *et al*. 2000). Éstas son:

1. La frecuencia total de ocurrencias en el texto.
2. La frecuencia de la multipalabra candidata contenida dentro de otra con mayor número de palabras.
3. El número de palabras de la multipalabra.

La frecuencia generalmente produce buenos resultados debido a que los términos tienden a ocurrir con relativa frecuencia. Sin embargo, una ventaja característica del *C-value* es que no sólo toma en cuenta la frecuencia de la multipalabra, sino también la frecuencia en la que ésta se encuentra anidada, es decir, contenida en otras.

Para calcular el C-value, se distinguen dos casos

1. Para las multipalabras que no se encuentran anidadas a otras, su "terminidad" será el resultado de su frecuencia y el número de palabras que contenga.
2. Para las multipalabras que se encuentran anidadas a otras, su "terminidad" se verá afectada por el número de veces en que ocurre como subconjunto.

$$C-value \begin{cases} \log_2|a| \cdot f(a) & 1 \\ \log_2|a|(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)) & 2 \end{cases}$$

Donde *a* es la multipalabra, *b* es la palabra que contiene *a*, *f(a)* es la frecuencia de *a* en el texto, *T* es el conjunto de multipalabras que contienen anidada a la multipalabra *a*, *P(T)* es el número de multipalabras del conjunto *T*.

Índice de gramaticalidad

La idea del índice de gramaticalidad se basa en el hecho de que las palabras funcionales tienden a ser, en general, palabras más cortas que las palabras de contenido; además de que las palabras funcionales, debido a su naturaleza, son mucho más frecuentes que las de contenido en cualquier documento.

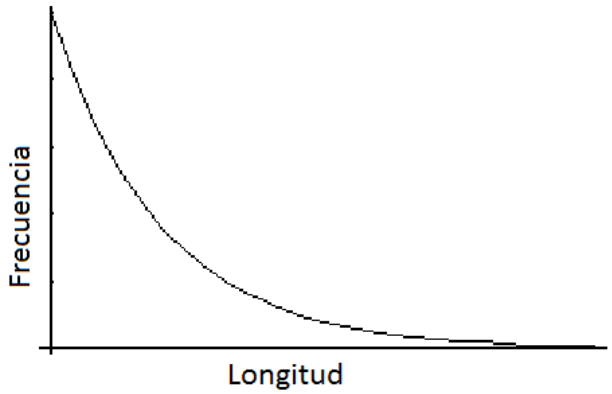

Figura 1. Relación entre la frecuencia y longitud de las palabras

Esto quiere decir que las palabras de menor longitud tienden a tener una mayor frecuencia dentro de cualquier documento lo que da una pauta para separarlas de forma automática sin el uso de listas de palabras funcionales.

De forma práctica, al graficar una lista de palabras de un documento se obtiene lo siguiente:

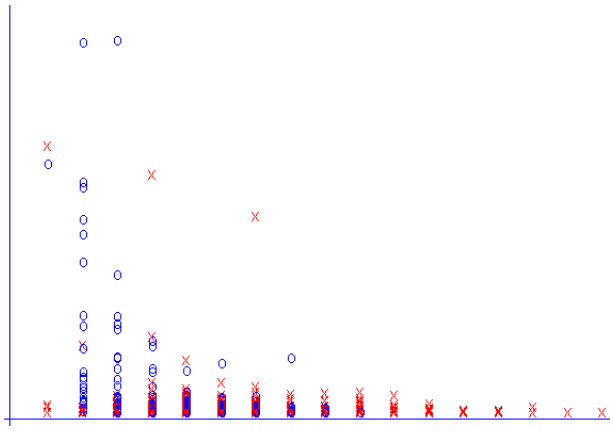

Figura 2. Lista de palabras de un documento

Donde los elementos graficados como un círculo son palabras que fueron identificadas como palabras funcionales por medio de la comparación con una lista de paro, el resto (marcadas con X), son palabras que no se encuentran en la lista de paro y, por lo tanto, se consideran como palabras de contenido.

En la gráfica anterior, pueden notarse 3 áreas importantes: el área dominada por los círculos, que son las palabras con menor longitud y mayor frecuencia, el área donde se encuentran mezclados los círculos y las equis en proporciones similares y el área dominada por las equis, que son palabras de mayor longitud y menor frecuencia.

Concentrándonos en la segunda zona, es fácil predecir que una separación exacta es imposible, tomando en cuenta que muchas palabras función no ocurren lo suficiente como para separarse del grupo de las palabras contenido, pero al concentrarnos en las zonas restantes se nota que sí existe una diferencia significativa, al menos, entre las palabras funcionales más comunes y las palabras de contenido más largas.

Tomando en cuenta lo anterior, se decidió hacer un programa que fuera capaz de dibujar una línea que separara, lo mejor posible, el grupo de palabras funcionales del de las palabras de contenido; y, con el objetivo de que dicha línea se

ajustara de acuerdo a los datos, se decidió recurrir al diseño de un *perceptrón* simple, cuyo funcionamiento se explica en el Anexo A (página 81).

El perceptrón se entrena usando los datos obtenidos de cada palabra en un documento hasta obtener los valores de una recta que se ajuste a los datos. Sin embargo, el usar dicha recta para separar arbitrariamente las palabras podría resultar en una clasificación errónea de las palabras cuyos datos las ubican más cerca de la frontera dibujada por dicha recta. Como se muestra en las siguientes gráficas:

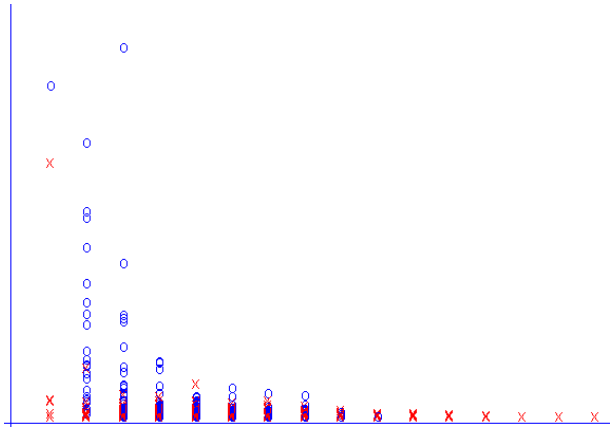

Figura 3. Palabras de un corpus separadas por una lista de paro

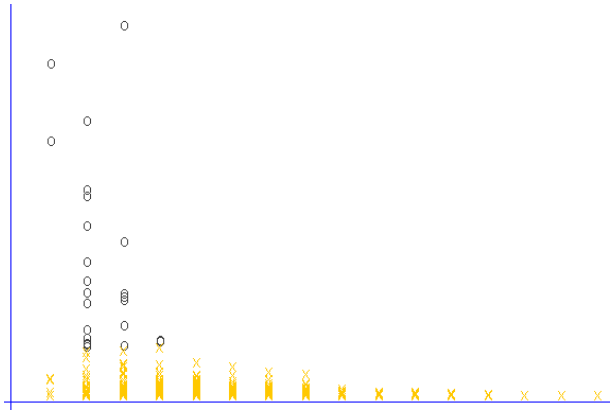

Figura 4. Palabras del mismo corpus separadas por recta

Como esta recta separa arbitrariamente los dos tipos de palabra, se decidió generar un índice que indique qué tanto una palabra dentro del documento se comporta o no como una palabra función. Para esto, se usó la función de transferencia *logsig* (logarítmica sigmoidal) en el perceptrón, lo que permitió generar un cociente entre 0 y 1 para cada palabra dentro del documento, donde el 0 indica que se trata de una palabra contenido y el 1 indica que se trata de una palabra función.

Este índice, denominado índice de gramaticalidad, tiene como objetivo el crear una generalización que pueda ser extrapolada a otros documentos y permita identificar palabras funcionales y de contenido sin la necesidad de hacer uso de listas de paro que se utiliza en gran parte de los desarrollos de herramientas de software de procesamiento de texto, aunque son costosas al tener que comparar cada token de un documento con cada elemento de dicha lista.

Otros métodos

Además de los métodos descritos anteriormente, podemos hacer mención del C-value/NC-value, cuya descripción se ofrece a continuación:

C-value/NC-value

El método C-value/NC-value (Frantzi *et al*, 2000) combina información lingüística y estadística para la extracción de términos. La primera parte, C-value, mejora la medida de frecuencia de ocurrencia de términos, que se utiliza generalmente, haciendo que sea sensible a términos multipalabra que se encuentran contenidos en otros, es decir, que se encuentran anidados. La segunda parte, NC-value, se obtiene tras realizar dos pasos: primero se utiliza un método para obtener palabras contextuales (palabras que tienden a ocurrir junto a los términos), el segundo consiste en la incorporación de la información de las palabras contextuales para la extracción de los términos.

El análisis lingüístico consiste en lo siguiente:

1. Se realiza un etiquetado del documento de categorías gramaticales (POS, *Part-of-Speech*). El etiquetado asigna una etiqueta gramatical (sustantivo, adjetivo, preposición, etcétera) a cada palabra del documento. Esta información es útil para el filtro.

2. Se filtra el documento etiquetado para remover palabras que no son requeridas para la extracción (verbos, adverbios, conjunciones, etc.). Por ejemplo, los términos suelen estar formados principalmente por sustantivos y adjetivos, en algunas ocasiones acompañados de preposiciones. Los filtros siguientes serían para la lengua inglesa (Frantzi *et al*, 2000, p.2):
    a) $Sustantivo^{+}Sustantivo$ (dos o más sustantivos)
    b) $(Adjetivo|Sustantivo)^{+}Sustantivo$ (serie de sustantivos o adjetivos con sustantivo final)
    c) $((Adjetivo|Sustantivo)^{+}|((Adjetivo|Sustantivo)^{*}$

$(SustantivoPreposición)^{?})(Adjetivo|Sustantivo)^{*})Sustantivo$
(serie de adjetivos o sustantivos o al menos un adjetivo o sustantivo seguidos de un sustantivo y preposición, etc.)

3. Se utiliza una lista de "paro" (*stop list*). Esta lista contiene palabras que no se espera que ocurran como términos o parte de ellos de cierto tema específico. Se utiliza para mejorar la precisión de la lista resultante al no extraer estas palabras (algún, la, los, no, muy, etc.).

Cálculo del C-value

El cálculo del C-value ha sido descrito anteriormente, por lo que sólo se explicará cómo se calcula el NC-value

Cálculo de NC-value

Este valor considera no sólo la frecuencia, longitud y relaciones entre los términos, sino que además toma en cuenta el contexto en el que éstos se encuentran. El NC-value se calcula de la siguiente manera para una secuencia de palabras *a*, cada una de las cuales (b_i) tiene un peso calculado en función de cuántas otras secuencias (a_j) son candidatas a término:

$$NC - value(a) = 0.8 \cdot CValue(a) + 0.2 \sum_{b \in C_a} f_a(b) \cdot weight(b)$$

$$weight(w) = \frac{t(w)}{n}$$

donde *b* es una palabra (por ejemplo, verbo, nombre o adjetivo) que aparece dentro del contexto del término. El peso *weight(w)* se calcula previamente, donde *t(w)* es la cantidad de candidatos con los que *w* aparece y *n* es el número de candidatos considerados $|a_j|$. Finalmente, $f_a(b)$ es el número de veces que la palabra *b* aparece dentro del contexto del candidato *a*.

Capítulo 3. Marco práctico

En este capítulo presentaremos algunas características del entorno que hemos decidido utilizar para el desarrollo de esta tesis. Esto es importante porque sienta las bases para el desarrollo, lo guía y conforma en el paradigma orientado a objetos.

Programación orientada a objetos

Antes de la creación del paradigma de la *programación orientada a objetos,* el paradigma dominante era el procedural, tal paradigma consiste en tratar de resolver los problemas al dividirlo en un conjunto de procedimientos, formados a su vez por tareas más pequeñas, donde el sistema resultante funcionaba como un flujo de procesos.

La programación orientada a objetos, por su parte, se centra en el modelado, a partir de problemas, de objetos y sus relaciones, con el objetivo de generar un sistema basado en dichos objetos y dichas relaciones.

> […] una aplicación desarrollada usando la programación orientada a objetos resultará en la producción de un sistema de cómputo que tiene una representación más cercana al dominio de problema del mundo real que si se hubiera usado la aproximación de programación procedural (Poo, Kiong, Ashok, 2007, p. 1; traducción nuestra).

Objetos y clases

Los *objetos* se construyen a partir de la abstracción de objetos que existen en el dominio del mundo real del problema. Tal abstracción se denomina *clase*. Una clase representa las propiedades y procedimientos que pertenecen al objeto real; comúnmente se define a la clase como un molde, a partir del cual se generarán uno o varios objetos. Wu (2010) lo describe de la siguiente forma: “un objeto es

una cosa, tan tangible como intangible que podemos imaginar" (p. 16; traducción nuestra).

Al proceso de generar un objeto de cierta clase se le denomina *instanciación,* por lo que, los objetos, también son llamados *instancias de clase* o simplemente instancias. Estas instancias son estructuras de datos que contienen las características descritas por la clase a la que pertenecen.

Atributos y métodos

Una clase puede definir dos tipos de características, atributos y métodos. Los atributos son elementos que integran al objeto y pueden ser tanto valores numéricos como textuales, o incluso otros objetos. Por otra parte, los métodos son conjuntos de instrucciones que ejecutan los objetos o las clases para realizar una tarea. Tal definición crea un paralelismo entre los métodos y las funciones en el paradigma procedural.

Para que un método se ejecute es necesario el uso de un mensaje que se genere desde otro método. Tal mensaje se denomina *invocación*. Muy parecida a una llamada a función, la invocación de un método contiene el nombre de la función y los parámetros que requiere para funcionar.

Herencia

Una de las ventajas de la *programación orientada a objetos* es el uso de la herencia de clases. Como lo define Wu (2010), "usamos un mecanismo llamado herencia para diseñar dos o más entidades que son diferentes pero contienen muchos atributos en común" (p. 23; traducción nuestra). Esto quiere decir que podemos generar subclases a partir de clases principales, donde cada subclase hereda los métodos y atributos de su clase padre y, además, es capaz de definir sus propios métodos y atributos, así como redefinir los heredados.

Cuando se redefine un método heredado, se dice que se realiza un *overide,* lo que permite modificar el comportamiento de un método existente desde la clase

padre, donde cabe destacar que el nombre y parámetros del método no se alteran, sólo las instrucciones internas.

Polimorfismo

Normalmente se cree que todo el código ha quedado *ligado* una vez que se ha realizado la compilación (*ligado estático*), pero el paradigma orientado a objetos permite un ligado en tiempo de ejecución (*ligado dinámico*). El ligado dinámico es bastante útil cuando se trabaja con herencia de clases, donde distintas subclases pueden tener distintas formas de realizar un mismo procedimiento, de acuerdo con las características específicas de cada subclase.

Como se describió, la herencia de clases permite modificar el comportamiento de los métodos heredados, pero cuando se tienen instancias de distintas subclases con una misma clase padre es difícil, para el compilador, determinar cuál de los métodos es el que se desea ejecutar.

La habilidad de distintos objetos de ejecutar distintos métodos a partir del mismo mensaje es conocido como *polimorfismo*. El polimorfismo indica que la interpretación de un mensaje no puede ser influenciada por el remitente, es decir que el mensaje de invocación de un método es enviado con el entendido de que existe un objeto que lo va a ejecutar, esto permite una gran flexibilidad en el uso de la herencia de clases y mayores facilidades para ampliar el código.

Lenguaje de programación Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Este lenguaje toma gran parte de su sintaxis de C y C++, con un manejo de clases más simple y elimina las instrucciones de bajo nivel.

Esto es, el lenguaje de programación Java está relacionado con C y C++ pero está organizado de forma diferente, con un número de aspectos de C y C++ omitidos y algunas ideas de otros lenguajes incluidas. “Está ideado para ser un

lenguaje de producción, no un lenguaje de investigación […]" (Gosling *et al*., 2011, p. 1; traducción nuestra).

Una de las características principales de este lenguaje de programación es que su compilación es *byte code*. Esto es, a diferencia de otros lenguajes donde la compilación genera un código de ejecución nativo, Java genera un código intermedio que es interpretado por su máquina virtual.

Máquina Virtual

El lenguaje de programación Java fue diseñado para ser multiplataforma, es decir, ser capaz de funcionar en distintas arquitecturas y sistemas operativos. Para que esto sea posible, el código de ejecución debe transportarse intacto entre las distintas plataformas, lo que presenta una dificultad si se desea que el software se ejecute de forma segura; ante esta dificultad, el lenguaje se creó para ejecutarse en un entorno de máquina virtual, donde se diseñaría una máquina virtual para cada plataforma y sobre estas máquinas se correrían todos los programas del lenguaje Java.

La máquina virtual de Java es la piedra angular de la plataforma Java, "es el componente de tecnología responsable por su independencia de hardware y de sistema operativo […] es una máquina de cómputo abstracta. Como una computadora real, tiene un conjunto de instrucciones y manipula distintas áreas de memoria en tiempo de ejecución" (Lindholm *et al*., 2011, p. 2; traducción nuestra).

En otras palabras, la máquina virtual es una computadora simulada dentro de otra computadora. Cada vez que se ejecuta código Java, se manda un conjunto de instrucciones a la máquina virtual de Java, instalada dentro de cada plataforma, donde ésta lo traduce a un conjunto de instrucciones nativas para la plataforma específica donde se ejecuta el código.

La clase *String*

Un *string* o cadena es una secuencia de caracteres que se manejan como una unidad. Para el manejo de cadenas, el lenguaje de programación Java cuenta con

una clase denominada la clase *String*. Lo que quiere decir que cada cadena en el programa es definida como un objeto de esta clase.

La clase *String* contiene un conjunto de métodos y atributos que facilitan la manipulación de cadenas de texto, herramientas que van desde el manejo de mayúsculas y minúsculas, sustitución de caracteres, hasta la comparación y subdivisión de cadenas. Estas herramientas son de mucha utilidad cuando se intentan resolver problemas relacionados con la lectura de documentos, o problemas de minería de textos.

Observaciones generales

El lenguaje Java es un lenguaje de programación multiplataforma y orientado a objetos, ambas características dotan a la aplicación Java de un gran potencial. Por un lado, la opción de ejecutar código de Java en distintas plataformas hace a la aplicación más útil y flexible; por otro lado, el uso del paradigma orientado a objetos permite modelar de forma más objetiva el dominio del problema, lo que facilita su uso para resolver problemas de la minería de textos, donde los problemas representan desafíos que van más allá de la lógica que plantea la programación procedural. En este capítulo, hemos presentado algunas características de la programación orientada a objetos y del lenguaje Java que nos ayudan a justificar utilizarlas en el desarrollo de esta tesis. En el capítulo siguiente, mostraremos aspectos específicos del desarrollo del mismo.

Capítulo 4. Desarrollo de un extractor de palabras

En este capítulo se muestra detalladamente cómo se procesa la información contenida en los documentos, con el fin de establecer el conjunto de palabras que lo represente de una mejor forma. Es decir, primero se procesa el texto y después, por medios estadísticos, se busca encontrar información relevante.

Se explican los procesos de: guardado del documento, identificación del idioma, generación de n-gramas, determinación de relevancia de expresiones y elección del conjunto de expresiones multipalabra más representativo del documento.

Procedimiento general

Para lograr la extracción de palabras clave de un documento, se debe analizar la información que éste contiene. Dicho análisis requiere que el texto se halle en un formato apropiado, para que su manipulación pueda realizarse de una manera más simple y menos compleja (computacionalmente hablando).

A continuación mencionamos los procedimientos que decidimos realizar para la extracción de términos, mostrando su obtención, análisis y presentación:

1. Preprocesamiento del texto

   En esta etapa se lleva a cabo el tratamiento del texto buscando que éste quede de una forma menos desestructurada mediante su representación en n-gramas lematizados.

2. Análisis de la información

   Es aquí donde se realiza el cálculo de la relevancia de cada *n-grama* determinado en la etapa anterior. Además, se establecen los criterios para la elección del conjunto final de palabras clave.

3. Presentación de los resultados

   Para que se muestre al usuario el conjunto de palabras clave del documento, se tiene que realizar un proceso de reconstrucción debido a que las palabras se encuentran lematizadas.

1. Preprocesamiento del texto

En este trabajo para procesar los documentos se siguieron las siguientes fases, que son descritas a continuación:

1. Se guarda el documento original
2. El documento es separado en enunciados
3. Se identifica el idioma del texto
4. Se generan y lematizan los n-gramas

Guardado del documento original

Al momento de que el usuario elige un documento, éste se guarda tal y como está. Esto se realiza con el fin de conservar las palabras en su forma original, ya que al realizar el proceso de lematización muchas de ellas serán reducidas a una expresión en común, pese a no tener relación semántica. De esta manera se puede obtener la palabra original. Otro motivo es el hacer más eficiente el cómputo de las operaciones, debido a que es más rápido hacer la lectura desde memoria principal que hacerlo desde el disco duro.

Separación del documento en enunciados

Para facilitar el manejo del texto, se dividió en enunciados. Consideramos un enunciado como aquello que se encuentra entre un punto y otro. El dividir el texto en enunciados facilita el proceso de textos pequeños ya que podemos considerar cada enunciado como un documento al momento de realizar el cálculo del TF-IDF para determinar la relevancia de las expresiones multipalabra.

Identificación del idioma

Para identificar el idioma del texto se utilizan las primeras líneas del documento y se buscan palabras características de cada idioma. Como generalmente un idioma varía mucho de otro, lo que se buscó fueron palabras funcionales, es decir: preposiciones, artículos, pronombres, verbos auxiliares, entre otros.

Una vez finalizado este procedimiento, el idioma con más coincidencias se establece como el del texto, si hay más de un idioma que comparte el número de coincidencias, se analizan más líneas hasta que exista una diferencia.

Generación de n-gramas

Después de procesar el texto, el programa extrae todas las posibles expresiones multipalabra que se encuentran en el documento. Estas expresiones son *n-gramas* y, en este caso, están formadas desde una hasta cinco palabras.

Antes de separar las palabras gráficas y reagruparlas en *n-gramas* es necesario depurar el documento para asegurarnos que las palabras gráficas obtenidas sean fácilmente identificables. Es por eso que se separan los signos de puntuación de las palabras por medio de espacios. Esto nos sirve para poder usar los signos de puntuación como límites de frase que serán validados por métodos que se explicarán más adelante, el siguiente fragmento muestra un ejemplo del resultado de esta separación:

> *volumes, pathogen concentrations, and dose–response parameters into the model. Median GI illness*

Después del proceso de separación, quedaría de la siguiente forma:

> *volumes , pathogen concentrations , and response parameters into the model . Median GI illness*

Posteriormente, se realiza la división del texto en palabras gráficas, donde se utiliza un conjunto de caracteres especiales para realizar la división, siendo el más importante el espacio simple pero incluyendo a otros símbolos como los sig-

nos de interrogación y exclamación, paréntesis, corchetes y otros. Éstos son delimitadores que se usarán para separar todas las palabras gráficas, creando un arreglo de cadenas de caracteres a partir de las cuales se generarán las listas de *n-gramas* del documento.

Clase *Enegrama*

Para almacenar *n-gramas* se creó una clase denominada *Enegrama* que contiene todos los atributos deseados del *n-grama*, como se muestra en la siguiente figura

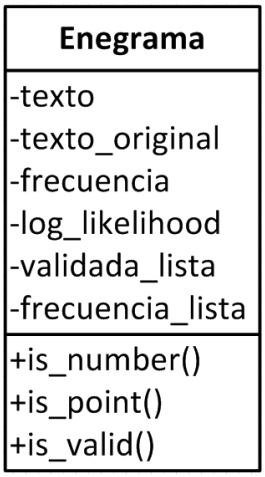

Figura 5. Diagrama de clase *Enegrama*

Donde podemos enumerar los siguientes atributos:

- *texto*: Conjunto de palabras gráficas lematizadas que representan al *n-grama*.
- *texto_original:* Cadena de palabras gráficas, resultado de la concatenación de los términos originales, previo a la lematización, que forman al *n-grama*.
- *frecuencia:* Número total de coincidencias del mismo *n-grama* en todo el documento.
- *log_likelihood, AM, TF_IDF:* Valores numéricos calculados sobre cada *n-grama.*
- *Validada_Lista:* Valor booleano que valida que el término aparezca en la lista de palabras clave del tema.

- *Frecuencia_Lista:* Su frecuencia en la lista de palabras clave, si es que aparece.

Por otra parte el método de validación *Is_Valid()* utiliza los otros dos métodos de la clase para determinar que ninguno de los elementos del n-grama sea un número o un signo de puntuación. Ambas condiciones crearían una ruptura de la expresión que, de acuerdo a nuestra propia consideración, anularía las posibilidades de que dicho n-grama sea una palabra clave.

Clase *ListaEnegramas*

Con el objetivo de estructurar la extracción de los n-gramas, cinco listas son necesarias para guardar los n-gramas del documento, donde cada una almacena un orden de n-grama que va desde el *unigrama* (una palabra gráfica), hasta el denominado *pentagrama* (términos de cinco palabras gráficas), después de los procesos correspondientes, las listas convergerán en una lista general de n-gramas, de la que hablaremos más adelante.

Para el almacenamiento de los *n-gramas* se creó la clase de *ListaEnegramas* la cual permite organizar todos los n-gramas del documento en una estructura de datos tipo lista ligada que hereda sus atributos de la clase *LinkedList,* clase propia de Java, y que tiene la siguiente estructura:

Figura 6. Diagrama de clase *ListaEnegrama*

De los atributos de dicha clase, el orden indica el orden de n-gramas almacenados en la lista y el atributo Stm es un objeto de la clase *Stemmer* responsable de aplicar el algoritmo de Porter.

Sobre los métodos de la clase, el método *agrega(),* permite agregar un *n-grama* a la lista, ya sea que ya exista o se agregue uno nuevo a la lista, el proceso se puede representar de la siguiente manera:

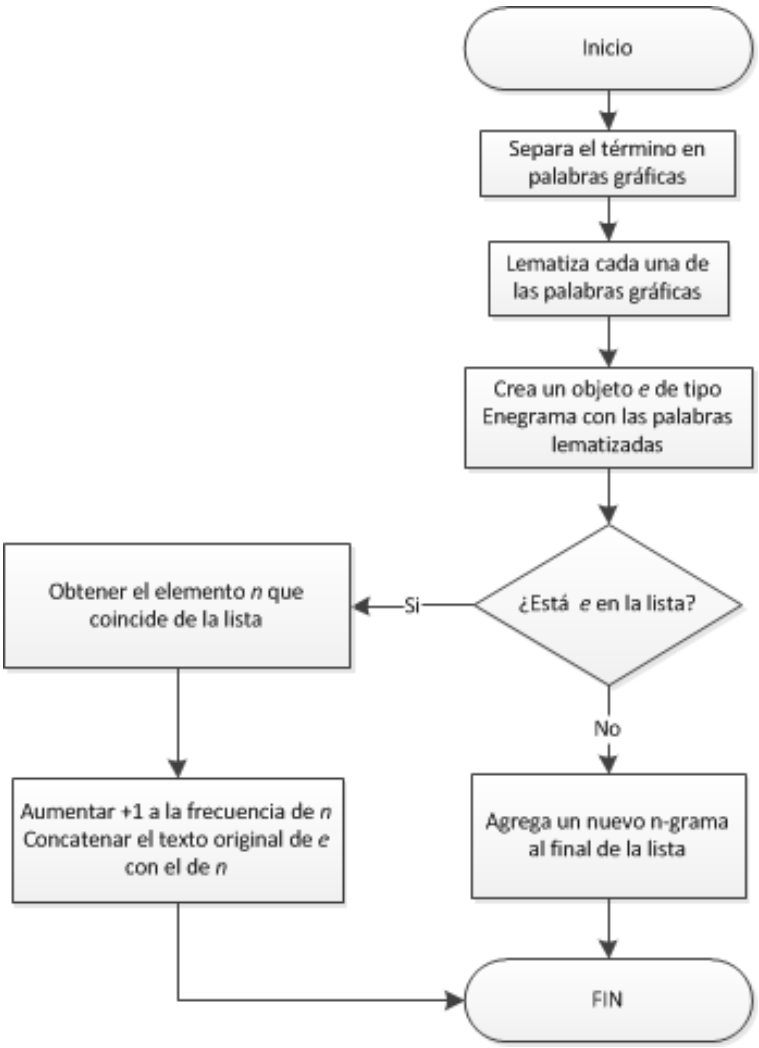

Figura 7. Diagrama de flujo. Método *agrega*

Como se puede ver, cuando el elemento que se desea agregar a la lista ya existe en la misma, el proceso se altera para aumentar la frecuencia de dicho término en el documento y para almacenar el texto original que generó al nuevo término. Esto se hace con el objetivo de reconstruir el texto original una vez que se ha finalizado con el procesamiento del documento y se desean mostrar los resultados.

Generación de unigramas

Los *unigramas* son los *n-gramas* formados por una sola palabra gráfica. Éstos se extraen directamente del arreglo de palabras del texto. Al estar formados por un solo elemento, no se puede calcular la razón de semejanza ni la medida de asociación AM, ni el C-value de los *unigramas*, lo que deja al *TF-IPF* y al índice de gramaticalidad como las únicas medidas de relevancia calculadas para unigramas.

La extracción de los *unigramas* es un proceso relativamente simple que consiste en agregar a la lista cada elemento que se obtiene del arreglo de palabras del documento, donde, como ya se vio en la figura 3, la palabra será lematizada y almacenada en la lista dependiendo de su presencia previa en la misma.

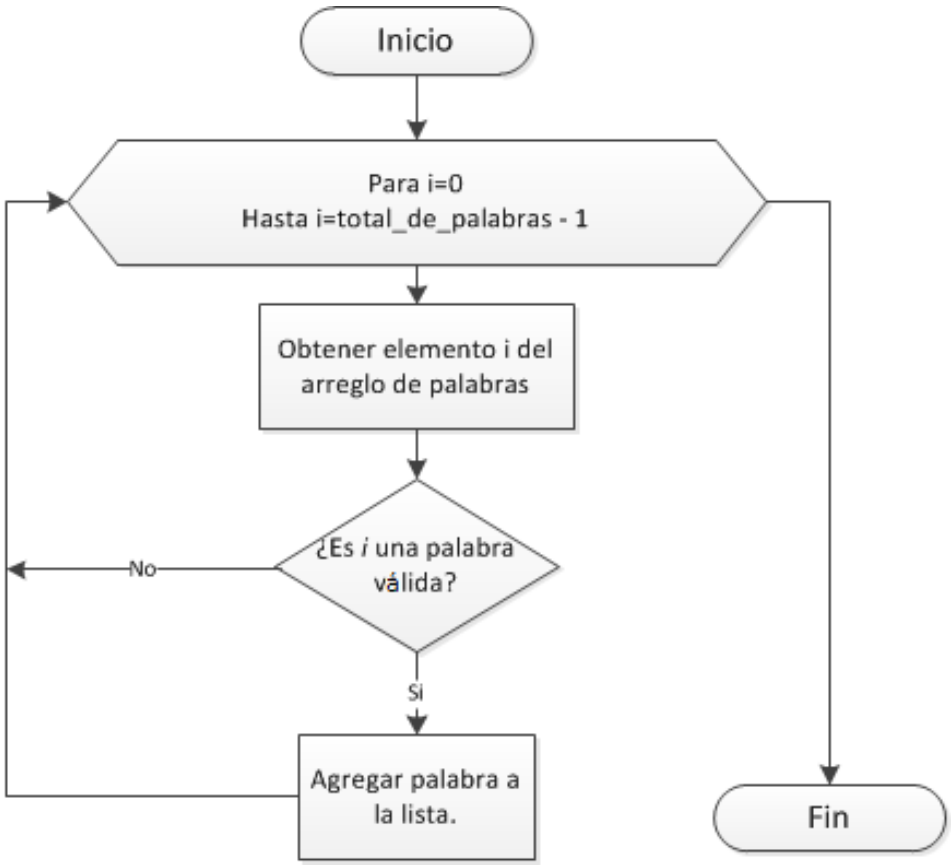

Figura 8. Diagrama de flujo. Extracción de unigramas

La validación de la palabra se realiza con el método *Is_Valid*(), descrito en la clase *Enegrama* y al agregar la palabra a la lista se hace uso del método *add*() de la clase *ListaEnegrama*.

Generación de bigramas

Los *bigramas* son términos formados por dos palabras gráficas que se encuentran de forma consecutiva en el documento. La extracción de *bigramas* se realiza dentro de un ciclo que extrae elementos del arreglo de palabras del documento en grupos de dos. Este par de elementos son unidos en una sola cadena, separados por un espacio, para ser agregados en la lista. El proceso es similar a la extracción de *unigramas*, con la diferencia de hacer una doble validación, una por cada elemento, además de realizar los cálculos de las medidas de similitud y asociación al final.

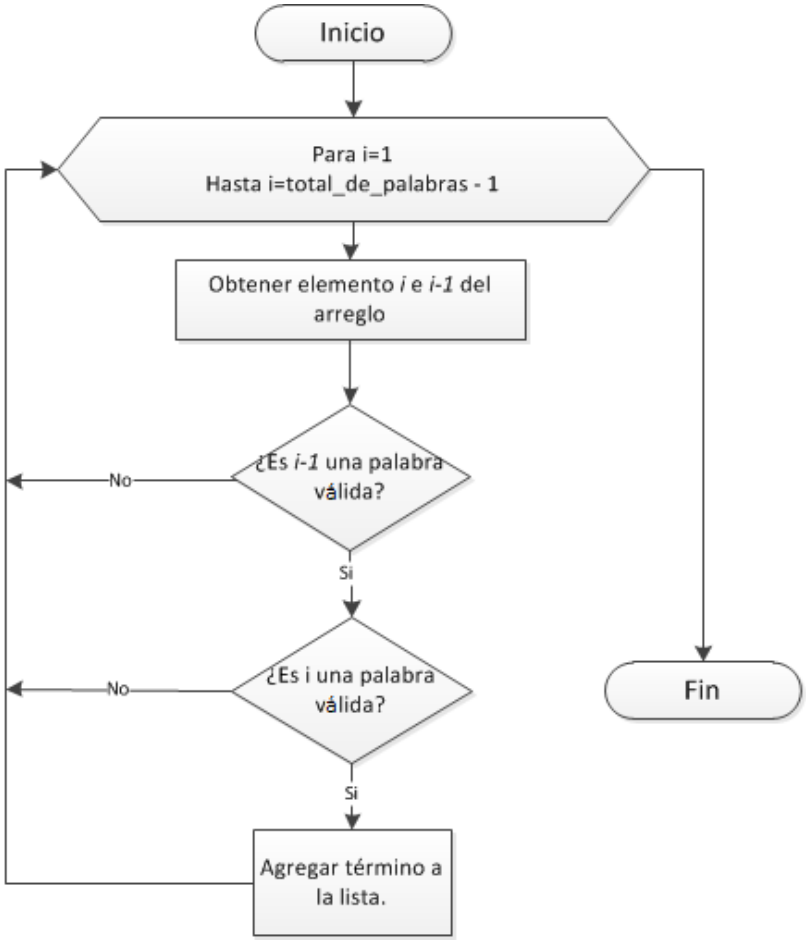

Figura 9. Diagrama de flujo. Extracción de bigramas

La validación se lleva a cabo en cada una de las palabras de la expresión para verificar que ninguna palabra sea un número o un signo de puntuación.

A partir de este proceso se obtiene una lista de expresiones con dos palabras gráficas; así que es posible, ahora, calcular un grupo de medidas como la medida de asociación AM y la razón de semejanza en un proceso que será descrito más adelante. Es importante resaltar esta parte del procedimiento ya que la generación de las listas posteriores depende de la forma en que la lista de *bigramas* sea organizada de acuerdo con estos cálculos.

Generación de n-gramas de orden tres y superiores

A partir de la obtención de la lista de *bigramas* y del cálculo de sus medidas de asociación (AM y razón de semejanza) es posible hacer un sesgo de las expresiones que nos pueden ser útiles para formar nuevos *n-gramas* de orden superior. Esto se hace al ordenar la lista de n-gramas de orden inferior de acuerdo a alguna de las medidas mencionadas (en este caso su asociación), y al cortar la parte inferior de la lista que se considere suficiente para omitir n-gramas poco asociados que tiene poca posibilidad de formar términos y menos aún términos clave en el documento.

Una vez que se ha acotado la lista, se puede formar una lista de *n-gramas* de orden superior al verificar que el conjunto de *n-1* palabras de una expresión (donde *n* es el orden de *n-grama* de la lista que se desea generar) pertenezcan a la lista de orden inferior. Esto para asegurar que el *n-grama* esté formado por un *n-grama* de orden inferior (n-1) suficientemente asociado para que la vinculación con una nueva palabra tenga altas probabilidades de ser un término.

El proceso anterior es igual para todos los *n-gramas* de orden superior a dos ya que es a partir del orden dos cuando se pueden calcular las medida de asociación. Estas medidas de asociación se deben calcular de nuevo considerando a cada nueva expresión como un tipo de *bigrama* formado ahora por un *n-grama* de orden inferior más una nueva palabra. Esta consideración permite aplicar el método del cálculo de las medidas de asociación que fue diseñado para los bigramas a la selección de los n-gramas de orden superior. El proceso detallado se muestra en el siguiente diagrama:

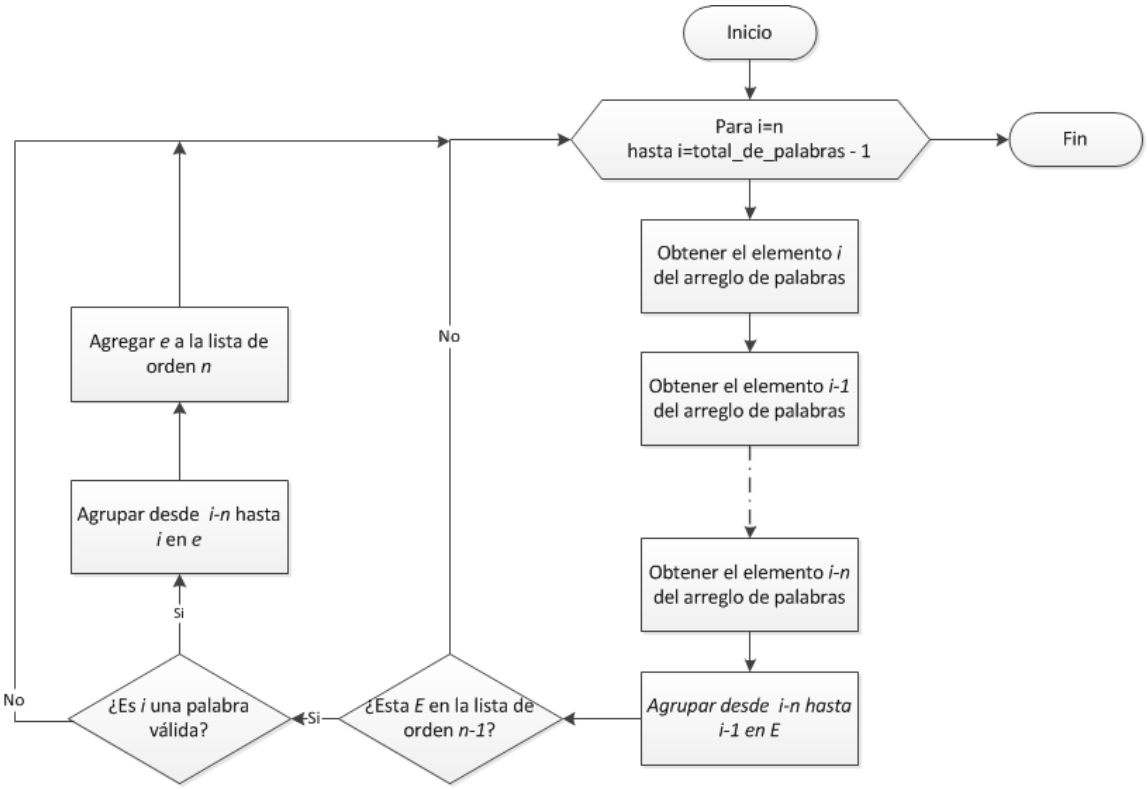

Figura 10. Diagrama de flujo. Generación de n-gramas de orden mayor a dos.

Este proceso se repite tres veces, desde *trigramas* hasta *pentagramas* (expresiones de cinco palabras gráficas). En cada una de las iteraciones, las listas de orden anterior son limitadas para asegurarnos tener elementos de mayor asociación formando las nuevas expresiones, así como reducir la cantidad de resultados que serán sometidos a los procesos posteriores.

Para poder realizar el acotamiento de las listas, es necesario recalcular las medidas de asociación entre cada iteración, además de ordenar las listas de acuerdo a los resultados obtenidos. Para esto, se ejecutan los métodos respectivos para cada una de las listas, de los que hablaremos a continuación.

Lematización del documento

Pese a que el algoritmo de Porter fue desarrollado originalmente para lematizar palabras escritas en inglés, existe un proyecto basado en éste y presenta la posibilidad de utilizarlo en más de dieciséis idiomas. Dicho proyecto se lleva a cabo en el marco del lenguaje de programación *Snowball,* el cual tiene implementaciones en C y Java y es el que utilizamos en este trabajo. Entonces, una vez que se identificó el idioma del texto se procede a emplear el algoritmo lematizador correspondiente.

Clase *Stemmer*

Esta clase contiene solamente un atributo, en el que se consigna el idioma que se quiere lematizar, los demás atributos son propios del algoritmo de Porter (ver anexo B).

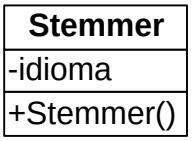

Figura 11. Diagrama de clase *Stemmer*

Donde

- *idioma*: indica mediante un número el idioma del texto: 1 para inglés y 2 para español.
- *Stemmer()*: recibe una palabra y la lematiza dependiendo del idioma.

Cabe señalar que el n-grama original se guarda en el atributo *texto_original* de la clase *Enegrama*. En caso de existir más de un n-grama que genere al mismo lema multipalabra, se añade también al atributo *texto_original*. Esto se hace con el fin de poder obtener la forma original del n-grama a partir de un n-grama lematizado.

2. Análisis de la información

Clase *Documento*

Para realizar el análisis de cada texto, se utilizan instancias de la clase *Documento*. Esta clase contiene tanto los atributos como los métodos necesarios para llevar a cabo el procesamiento completo del texto de un solo documento. Una vez que ya se tiene la lista de n-gramas lematizados, se procede a calcular la calificación de relevancia para cada uno, este proceso se realiza apoyándose en la clase *MedidasAsociación*, que se describe posteriormente.

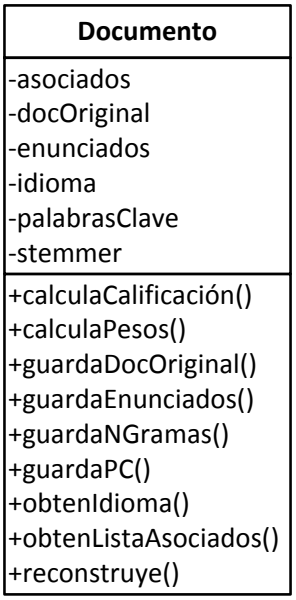

Figura 12. Diagrama de clase *Documento*

Se explican a continuación los atributos y métodos de la clase *Documento*:

Atributos:

- *asociados*: lista de n-gramas que será analizada.
- *docOriginal*: conjunto de palabras gráficas correspondientes al documento ingresado por el usuario sin ninguna modificación.

- *enunciados*: conjunto de secuencias de palabras gráficas contenidas en el texto entre un punto y otro.
- *idioma*: representación por medio de un número entero del idioma del texto.
- *stemmer*: instancia de la clase *Stemmer* encargada de realizar el proceso de lematización.
- *palabrasClave*: lista de palabras clave provenientes de un catálogo elegido por el usuario con el fin de apoyar la elección de las expresiones multipalabra características del documento.

Métodos:

- *calculaCalificacion*(): en caso de que haya sido ingresado por el usuario un conjunto de palabras con el fin de evaluar su consistencia, éste es el método encargado de asignarle una calificación.
- *calculaPesos*(): calcula tanto la razón de semejanza como la medida de asociación para cada n-grama.
- *guardaDocOriginal*(): guarda el documento original.
- *guardaEnunciados*(): guarda los enunciados del texto.
- *guardaNGramas*(): realiza el guardado de los n-gramas del texto.
- *guardaPC*(): en caso de haber sido seleccionada una colección de palabras clave por el usuario, guarda la lista de éstas.
- *obtenIdioma*(): determina el idioma del texto.
- *obtenListaAsociados*(): guarda una lista con los n-gramas que serán analizados.
- *reconstruye*(): recupera el n-grama original partiendo de uno lematizado.

Clase *MedidasAsociacion*

La clase *MedidasAsociacion* contiene todos los métodos necesarios para calcular los atributos de medición, como la medida AM, el C-value y la razón de semejanza a partir de la lista de n-gramas. Además en caso de que se utilice un archivo histórico aumenta la relevancia de una palabra contenida en él. La representación de la clase es la siguiente:

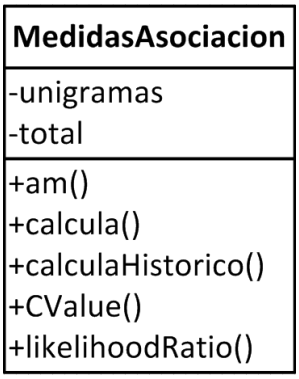

Figura 13. Diagrama de clase *MedidasAsociacion*

Esta clase, aunque sencilla a simple vista, contiene los métodos de análisis numérico y estadístico que determinan la relevancia de las expresiones como términos representativos del documento. Sus atributos son:

- *unigramas*: objeto de la clase *ListaEnegramas* que es la lista de los n-gramas sencillos que forman al documento.
- *total*: variable entera que representa el número de palabras gráficas del documento.

Ambos atributos son fundamentales para obtener las valores necesarios en el cálculo de los atributos del *n-grama* que analizamos en este trabajo. Además de esto, los métodos extraen la información faltante de cada lista, como se muestra a continuación:

Método *calculaHistorico()*

Calcula el valor tomando en cuenta los valores anteriores de frecuencia, asociación y razón de similitud, sólo en caso de que el n-grama aparezca en el archivo histórico.

Método *CValue()*

Este método calcula el CValue de un n-grama. Este método es invocado por la clase *ListaEnegramas*.

Método *calcula()*

Este método es un método que engloba la obtención de valores de las variables para el cálculo de la medida de asociación (AM) y la razón de similitud (*likelihood ratio*).

El método se apoya de otros métodos en la clase que hacen más flexible el procedimiento al tener que trabajar con listas de dos y hasta cinco elementos. El funcionamiento del método se muestra a continuación:

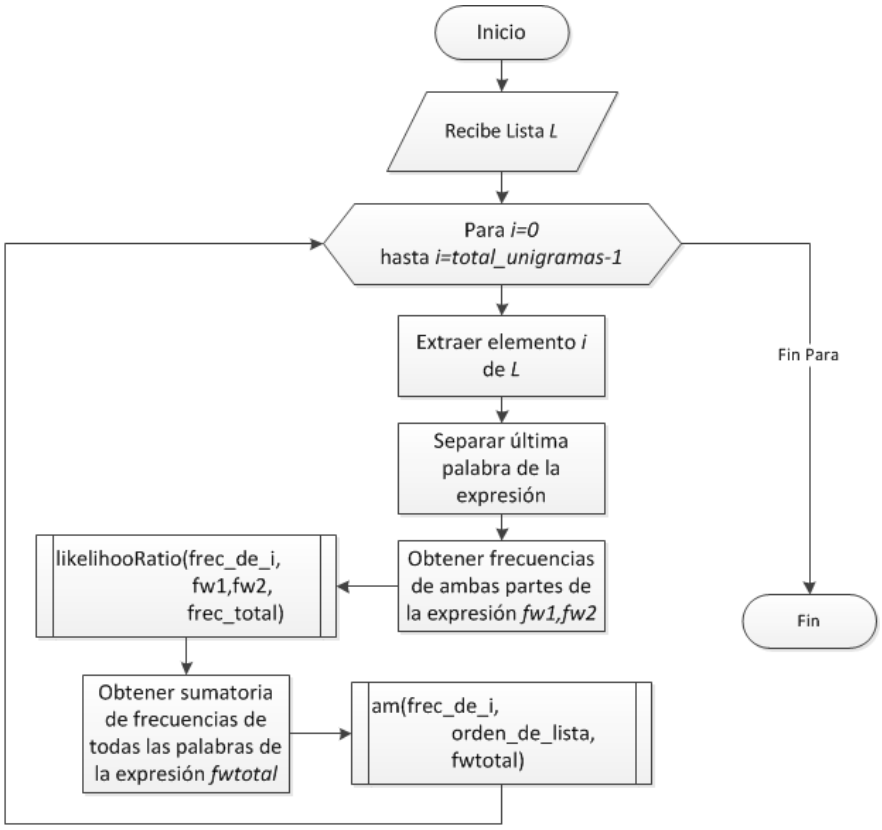

Figura 14. Diagrama de flujo. Método calcula

Como se puede ver en el diagrama, el método *calcula* necesita de información sobre las distintas palabras que forman el término en orden de poder calcular los distintos atributos definidos en la clase *Enegrama*. Es por eso que es necesario

crear un objeto de la clase *MedidasAsociacion* que contenga la lista de *unigramas* como atributo, de donde se extraerá la información de cada palabra.

A partir de este proceso, se obtienen los valores de las medidas de asociación vinculadas con cada *n-grama.* Al tener la listas completas, es momento de generar una lista general que contenga los elementos más asociados de cada una de las listas, para esto se realiza un proceso iterativo que consiste en concatenar cada una de las listas en una lista única, misma que será ordenada por medio de un proceso nativo de la clase *LinkedList* de Java. En el caso de los unigramas, se consideró asignarles la asociación máxima de la lista anterior, ya que los *unigramas* al estar formados por una sola palabra, no son sujetos a las fórmulas del cálculo de las medidas de asociación (y presumimos que no hay nada más asociado que lo que no está separado). Sin embargo, como se verá en el capítulo siguiente esta consideración no resultó útil por lo que se decidió omitir a los *unigramas* de los resultados finales[1].

El algoritmo de ordenamiento usado por Java es el algoritmo *merge sort* u ordenamiento por mezcla, basado en el algoritmo divide y vencerás. Su algoritmo se observa a continuación:

[1] De hecho, incluso intentamos multiplicar esta asociación máxima con el índice de gramaticalidad para no descartarlos, pero tampoco resultó satisfactorio.

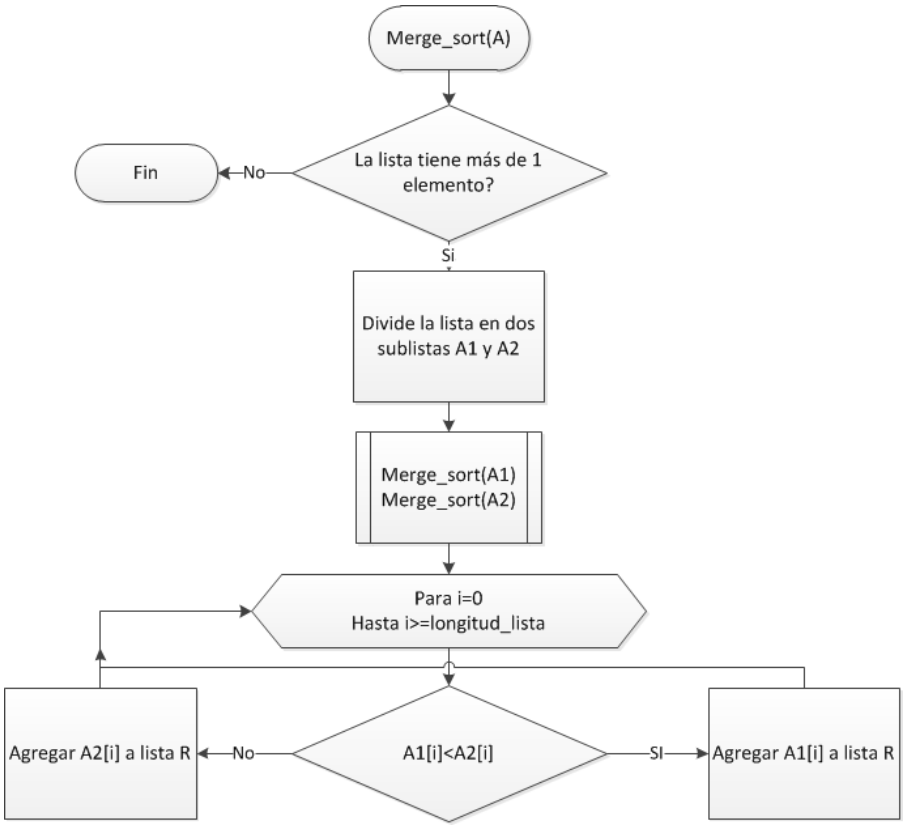

Figura 15. Diagrama de flujo. Ordenamiento por mezcla

3. Presentación de primeros resultados

Al finalizar el análisis de los n-gramas por medio del método *calcula()*, la aplicación ordena la lista de n-gramas de acuerdo a la calificación de relevancia obtenida de cada uno. Finalmente, se toman los 10 n-gramas con la mayor calificación para ser mostrados al usuario final como resultado del análisis del texto. Los n-gramas mostrados provienen del método *reconstruye()*, el cual toma un n-grama lematizado y muestra su forma original (la cual se conservó según describimos en la sección anterior: Preprocesamiento del texto – Guardado del documento original). Como un lema multipalabra puede provenir de diferentes n-gramas, se toma aquél con mayor frecuencia. En caso de que la frecuencia sea la misma entre dos o más n-gramas, se toma el n-grama que tenga la menor longitud, es decir, la menor cantidad de letras.

Almacenamiento de información histórica

Después de cada extracción de palabras clave, nuestra aplicación almacena los datos obtenidos para ser utilizados en futuras extracciones. De esta manera, se retoman datos estadísticos de n-gramas que se repiten en el procesamiento secuencial de documentos. Es evidente, que un registro de este tipo es necesario para cada área temática y cada idioma que se trabaje. De esta manera, podemos hablar de una especie de aprendizaje.

Específicamente, nuestra aplicación tiene la opción para indicar el tema del que trata un documento en específico. Este tema ayuda a categorizar a los documentos analizados creando un archivo al que llamamos *historia*. Así, este archivo se utiliza con el fin de obtener un conjunto de candidatos a término, no de un documento, sino del tema del que trata. Como ya se explicó, este proceso se realiza guardando los candidatos a término de documentos analizados previamente con un tema en común. De esta manera, la lista de candidatos a término del tema ayudará en análisis posteriores de documentos que compartan el tema, ya que con esta información se puede utilizar el TF-IDF que ha sido explicado en el marco teórico.

Capítulo 5. Pruebas realizadas

Como se menciona en el marco teórico, después de obtener el conjunto de n-gramas de un documento, comienza el proceso para determinar cuáles de ellos son los más representativos del documento en cuestión. Para obtener un valor de relevancia de los candidatos a vocablos multipalabra, tenemos:

Dado un candidato a multipalabra *a*:

- *LL(a)* es la razón de semejanza.
- *AM(a)* es la asociación entre las palabras que contiene.
- *TF-IPF(a)* es qué tan relevante es en el documento actual.
- *IG(a)* es su Índice de gramaticalidad.
- *CV(a)* es el *C-value* de a.
- *TF-IDF* es qué tan relevante es entre el conjunto de documentos pertenecientes a un mismo tema.

Con base en estas medidas se realizaron pruebas para determinar la eficacia de cada una, viendo si los resultados eran útiles para cumplir el objetivo del presente trabajo. La primera fase del experimento consistió en calcular cada una de manera individual. Para esta parte, se procesó el siguiente documento (Albuquerque, 2009):

Título: “Machos a la media luz: miradas de una antropología impropia”
Autor: Camilo Albuquerque de Braz
Institución: Universidad Estatal de Campinas, Brasil

Algunas consideraciones para las pruebas

Como estas pruebas se plantearon para un texto de sexualidad, área temática para la que no contamos con una lista predefinida de términos, omitimos el uso del atributo *Validada_Lista* que se describió en el capítulo 4. Además, debido a que la mayoría de las medidas utilizadas se basan en al menos dos unigramas para el cálculo de la calificación de relevancia de su asociación, los resultados encontrados al incluir a los unigramas, no resultaron útiles. En esencia, el principal problema fue encontrar una medida que determinara una jerarquía entre unigramas. Las únicas medidas con las que contamos son el Índice de Gramaticalidad y el TF-IPF y nos resultó difícil compararlas en aislamiento con las otras medidas que miden la asociación entre unigramas. De esta manera, los unigramas tendían a obtener calificaciones muy similares entre sí y por consiguiente aparecían agrupados en bloque, ya sea al final o al principio de la lista. En consecuencia, se optó por omitir a los unigramas en el cálculo final de relevancia.

En la tabla siguiente aparecen los 10 unigramas mejor evaluados. Nótese que como no recurrimos a un etiquetado de categorías gramaticales tenemos verbos, nombres propios y adverbios que son pobres candidatos a palabras clave; de allí nuestra decisión de omitirlos.

Unigramas
entrevisté
véase
macrae
foucault
distinción
barrio
gregori
muchos
rubin
gls

Tabla 2. Resultado tras analizar el documento especificado anteriormente.

Combinaciones propuestas

En la siguiente tabla se ven los resultados de nuestra aplicación.

	LL	AM	TfIpf	IG	C-value
1	sao paulo algunos clubes presuntamente	sao paulo	en el	antropología impropia	de la
2	sao paulo por lo menos	de la	de las	camilo albuquerque	en la
3	sao paulo contemporáneo	cine porno	de los	universidad estatal	de los
4	sao paulo algunos clubes	ciertos estereotipos	la sexualidad	diseño sustancializa	en el
5	sao paulo algunos	georges bataille	en los	aportar pruebas	de las
6	sao paulo por lo	glory holes	el género	pruebas empíricas	en los
7	sao paulo por	en la	sexual y	maría filomena	el género y la sexualidad
8	sao paulo	locales comerciales	la ciudad	violencia interpersonal	y la
9	de la avenida ipiranga macrae	ya sea	de sexo	mutuamente imbricadas	a la
10	de la violencia interpersonal	sin embargo	en la	occidentales contemporáneas	y bares de sexo

Tabla 3. Resultados tomando en cuenta sólo una medida

El análisis de cada medida de esta tabla, se muestra a continuación:

1. Razón de semejanza (primera columna)

   Los resultados están ordenados de mayor a menor asociación entre las palabras que contienen[2]. Se observa que las cadenas: "*sao paulo algunos clubes presuntamente"* y "*sao paulo por lo menos"* se encuentran cerca en la tabla porque ambos incluyen al digrama "*sao paulo"*. De los primeros 10 n-gramas podemos destacar *sao paulo contemporáneo* y *de la violencia in-*

[2] Como se apuntó en el marco teórico, la razón de semejanza calcula estadísticamente qué tanto se asocian dos entidades, en este caso n-gramas.

terpersonal, como ejemplos de sintagma nominal y sintagma preposicional bien formados[3], respectivamente.

2. Medida AM entre palabras (segunda columna)
   Esta lista muestra los n-gramas más asociados entre sí según la medida AM (Weiss *et al* 2005:34). Destacan: "*sao paulo"*, "*cine porno"*, "*ciertos estereotipos"*, "*georges bataille"* y "*locales comerciales"* como cadenas bien formadas.

3. TF-IPF, *total frequency-inverse paragraph frequency*, (tercera columna)
   Presenta como resultado los n-gramas que tienden a ocurrir más en un menor número de párrafos de un documento dado. Destacamos de la lista: "*la sexualidad"* y "*el género"*, que son sintagmas nominales bien formados, aunque seguramente candidatos pobres a ser palabras clave por incluir los artículos.

4. Índice de gramaticalidad (cuarta columna)
   Esta lista nos muestra n-gramas que no contienen palabras funcionales. Son resultaron interesantes: "*antropología impropia"*, "*camilo albuquerque"* y "*violencia interpersonal"*.

5. *C-value* (quinta columna)
   Dada la naturaleza de esta medida, los resultados tendrían que ser más extensos en cuanto al número de palabras por n-grama. No obstante, de los primeros 10 resultados sólo se ve bien formada la cadena: "*el género y la sexualidad"*, que tampoco parece buena candidata como palabra clave.

Como se ve, ninguna medida por sí sola parece capaz de entregar los resultados que se buscan. Por lo tanto, es necesario encontrar una forma de combi-

[3] Manifiestamente no buscamos seleccionar sintagmas preposicionales como palabras clave. Sin embargo, es de notarse que se trata de expresiones bien formadas, lo cual es ya un logro importante.

nar estas medidas para mejorar el resultado, originando una segunda fase de pruebas.

La segunda fase consistió en multiplicar todas las medidas exceptuando una a la vez. Los resultados se muestran a continuación:

	sin LL	**sin AM**	**sin Tflpf**	**sin IG**	**sin C-value**
1	el género y la	el género y la	el género y la	de la	sao paulo
2	sao paulo	sao paulo	sao paulo	de los	ciertos estereotipos
3	los clubes y bares	los clubes y bares	los clubes y bares	en la	georges bataille
4	georges bataille	en la producción	georges bataille	en el	glory holes
5	sin embargo	así como	sin embargo	sao paulo	estuviesen dispuestos
6	y simíµes	sin embargo	camas colectivas	en los	cine porno
7	camas colectivas	y simí**µ**es[4]	en la producción	de las	camas colectivas
8	aquellos cuya	georges bataille	así como	el género	locales comerciales
9	matriz cultural	punto de	cine porno	lo que	sin embargo
10	así como	camas colectivas	aquellos cuya	la sexualidad	aquellos cuya

Tabla 4. Resultados sin tomar en cuenta una medida

De nueva cuenta analizamos los primeros 10 resultados de cada combinación.

1. Sin Razón de semejanza

   Destacan como expresiones bien formadas: *sao paulo*, *los clubes y bares, georges bataille*, *camas colectivas*, y *matriz cultural*

[4] Representa un error al momento de hacer la codificación a texto plano, el texto original es y Simões, refiriéndose a un nombre propio.

2. Sin Asociación AM entre palabras
   Destacan: *sao paulo*, *los clubes y bares, georges bataille y camas colectivas.*

3. Sin TF-IPF
   Destacan: *sao paulo*, *los clubes y bares, georges bataille, camas colectivas* y *cine porno.*

4. Sin Índice de gramaticalidad
   Destacan: *sao paulo*, *el género* y *la sexualidad*.

5. Sin C-value
   Destacan: *sao paulo, ciertos estereotipos, georges bataille, cine porno, camas colectivas y locales comerciales.*

Además se realizó una prueba donde se multiplicaban las cinco medidas estadísticas. Destacando, por estar bien formadas, las expresiones: *sao paulo, los clubes y bares, georges bataille, camas colectivas y cine porno.*

	TODAS
1	el género y la
2	sao paulo
3	los clubes y bares
4	sin embargo
5	así como
6	en la producción
7	georges bataille
8	y simíμes
9	camas colectivas
10	cine porno

Tabla 5. Resultados tomando en cuenta las 5 medidas

Después de analizar los resultados de la segunda fase sobresalen las combinaciones: sin TF-IPF, sin C-value y la que implica a todas las medidas, porque presentan el mayor número de expresiones bien formadas. Conviene enfatizar que en general no se ven como buenas candidatas a palabras clave, pero consideramos que el que se vean bien formadas es un criterio de suma importancia.

Encuesta

Para apoyarnos en la elección de una combinación de estas medidas estadísticas, realizamos una encuesta donde mostramos los resultados de las combinaciones, tras analizar con nuestra aplicación dos documentos de carácter científico. El primero en español y segundo en inglés. Para el experimento en español se utilizó el documento mencionado anteriormente. Para el idioma inglés utilizamos un conjunto de resúmenes de documentos relacionados con el tratamiento de aguas residuales. Pedimos que seleccionaran las expresiones bien formadas que le ayudaran a inferir el tema del que trataba cada documento. Para consultar el formato de la encuesta consulte el Anexo C.

.

Capítulo 6. Evaluación

Resultados de la encuesta

Una vez que las encuestas fueron aplicadas, fue necesario diseñar una forma efectiva de cuantificar los resultados con el objeto de determinar cuál de los métodos utilizados para formar las listas es más eficiente para conseguir mostrar palabras claves.

Los datos se encontraban dentro de una tabla, donde cada encuestado había seleccionado los términos que le parecieran coherentes y bien formados dentro de cada lista. Con estos datos era importante que la lista tuviera la mayor cantidad de términos calificados de forma positiva, pero también era importante considerar la posición de los términos dentro de la lista para encontrar aquella que le dé más prioridad a los términos bien formados.

Se decidió usar los términos calificados de forma negativa para calcular el "error" dentro de cada lista, lo anterior se consiguió al usar tanto el número de errores de la lista como su posición dentro de la misma.

Para cada encuestado se tiene una tabla como la siguiente:

Lista	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	X				X		X	X		X		X		X	
2	X	X	X	X	X	X		X	X	X			X	X	X
3		X	X	X			X			X		X			
4		X		X	X		X		X	X	X		X	X	X
5	X	X		X	X	X		X		X	X	X	X	X	X
6	X	X	X	X		X	X		X		X		X	X	X
7					X				X	X			X		
8	X			X	X	X		X			X		X	X	X

Tabla 6. Tabla de captura de resultados

Donde cada fila representa una lista generada por el uso de una fórmula diferente y cada columna representa una expresión dentro de la lista, siendo la primera columna la expresión mejor calificada por nuestro método.

La tabla muestra con una ‘X’ todos aquellos términos en la lista que, de acuerdo con los encuestados, no representan un término bien formado o relevante.

Para calcular el grado de error de cada una de nuestras listas era necesario medir la cantidad de errores y su posición, para lograr esto se empleó un método sencillo que se explica a continuación.

En primer lugar se obtuvo el coeficiente para cada palabra a través de la división del número total de palabras entre la posición de cada uno de los errores. Un error en la primera posición obtiene un valor de 15 y un error en la última posición un valor de 1, obteniendo los términos correctos un valor de cero, tal como se muestra a continuación:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	15	0	0	0	3	0	2.143	1.875	0	1.5	0	1.25	0	1.071	0
2	15	7.5	5	3.75	3	2.5	0	1.875	1.667	1.5	0	0	1.154	1.071	1
3	0	7.5	5	3.75	0	0	2.143	0	0	1.5	0	1.25	0	0	0
4	0	7.5	0	3.75	3	0	2.143	0	1.667	1.5	1.364	0	1.154	1.071	1
5	15	7.5	0	3.75	3	2.5	0	1.875	0	1.5	1.364	1.25	1.154	1.071	1
6	15	7.5	5	3.75	0	2.5	2.143	0	1.667	0	1.364	0	1.154	1.071	1
7	0	0	0	0	3	0	0	0	1.667	1.5	0	0	1.154	0	0
8	15	0	0	3.75	3	2.5	0	1.875	0	0	1.364	0	1.154	1.071	1

Tabla 7. Tabla de calificación de resultados

De esta forma, cada una de las expresiones calificadas de forma negativa obtenía un valor de acuerdo a su posición. Los datos obtenidos se promediaron para cada una de las listas.

Después de procesar los datos para cada usuario se promedian los datos de las listas de todos los usuarios, obteniendo los siguientes datos:

	1	2	3	4	5	6	7	8
Promedio	1.82	2.89	1.50	1.42	2.55	2.77	0.48	2.01
Promedio normalizado	0.63	1.00	0.52	0.49	0.88	0.96	0.17	0.70
Complemento	0.37	0.00	0.48	0.51	0.12	0.04	0.83	0.30
Desviación estándar	0.22	0.19	0.47	0.20	0.36	0.19	0.13	0.22
Factores	**¬TF-IPF**	CV	G	AM	LL	**¬G**	**¬CV**	Todos

Tabla 8. Tabla con los resultados finales

La tabla 8, que ahora muestra en las columnas las fórmulas aplicadas, nos permite ver el grado de “éxito” de cada una de las listas de la encuesta, donde un valor mayor en el complemento del promedio indica un menor número de errores en las primeras posiciones de la lista. Como se dijo, en las columnas se especifican las fórmulas; por ejemplo, en la columna 1 se tiene la **negación** del factor TF-IPF (lo que quiere decir que la fórmula utilizó todos los factores excepto ese), mientras que en la columna 2 se muestran los resultados obtenidos únicamente al aplicar la fórmula **C-value**.

Cabe señalar que sólo se analizaron los resultados correspondientes al idioma español, debido a que el número de personas consultadas con conocimientos del idioma inglés fue reducido (aproximadamente el 30%) y, al ser hispanoparlantes, pudimos ver que su capacidad de determinar si una expresión del inglés está bien formada fue cuestionable.

Análisis de los resultados

A partir de estos resultados es posible determinar que la lista número 7 es la que tiene una mejor puntuación (el menor promedio de error con la menor desviación estándar), siendo esta lista la correspondiente con la fórmula que no usa el **C-value** para el cálculo de la relevancia de los términos. Esto no es una sorpresa, porque nuestra aplicación no etiqueta las categorías gramaticales, requisito importante para el cálculo del **C-value** (para quedarse sólo con expresiones bien formadas a priori).

Al ver el contraste entre las fórmulas mejor calificadas con las de menor calificación es posible evaluar las fortalezas y debilidades de nuestro método así como observar cómo cada uno de los factores obtienen una lista muy específica de términos que define su comportamiento y hace más fácil considerar su peso e impacto en el método final.

Los resultados obtenidos fueron bastante útiles para determinar los mejores elementos de nuestro programa para otorgar mayor relevancia a términos que, de forma general, se pueden definir como bien formados.

Por lo tanto, se utilizó la lista 7 para mostrar los resultados finales (ver tabla 9). A continuación se muestran los resultados tras analizar el artículo mencionado anteriormente.

	Nuestra aplicación
1	sao paulo
2	ciertos estereotipos
3	georges bataille
4	glory holes
5	estuviesen dispuestos
6	cine porno
7	camas colectivas
8	locales comerciales
9	sin embargo
10	aquellos cuya

Tabla 9. Resultados tras el análisis con nuestra aplicación

Como nuestra aplicación no presupone un etiquetado sintáctico ni una configuración gramatical previa de las expresiones clave que buscamos, tenemos entre los primeros 10 candidatos cadenas como “estuvieron dispuestos” y “aquellos cuya” que no son expresiones bien formadas ni pueden ser consideradas estructuras clave. Es significativo que locuciones como “sin embargo” ocurran entre estos primeros candidatos (más adelante ocurren “ya sea”, “así como”, etc.), puesto que son expresiones recurrentes estructuradoras del discurso en español (cabe esperar que serían eliminadas automáticamente al recurrir al almacenamiento histórico de los candidatos).

Comparación de los resultados con otras aplicaciones

Una vez establecida la función para determinar la relevancia de los candidatos a vocablos multipalabra, se procedió a comparar los resultados de nuestra aplicación con los del extractor terminológico TERMEXT[5] y con otras dos aplicaciones en línea pero de carácter comercial: (1) AlchemyAPI - Terminology Extraction[6] y (2) Translated Labs - Terminology Extraction[7].

Para la comparación se utilizó el mismo artículo que utilizamos en el capítulo anterior. Esto es, se compararon los resultados de este artículo (los primeros 10 candidatos a término) con cada una de las tres aplicaciones. A continuación presentamos información sobre TERMEXT, AlchemyAPI y Translated Labs.

[5]TERMEXT es un extractor terminológico basado en el método CN-value (Ananiadou y McNaught, 2006) descrito en el marco teórico, desarrollado en el Instituto de Ingeniería (Barrón-Cedeño *et al*, 2009): http://www.iling.unam.mx/termext/

[6] Http://www.alchemyapi.com/api/keyword/
[7] Http://labs.translated.net/terminology-extraction/

TERMEXT

Como se establece en la página de TERMEXT, éste “extrae automáticamente candidatos a término a partir de textos especializados de un texto plano”. El procedimiento consta de tres módulos:

- El módulo 1 realiza un etiquetado a nivel morfosintáctico.
- El módulo 2 calcula el NC-value para cada candidato a término y construye una lista de los mismos.
- El módulo 3 analiza la lista de candidatos a término mediante el NLTK (*Natural Language Toolkit*) de Python. Lo importante es que se basa en el patrón morfosintáctico, para ofrecer una lista definitiva de candidatos a término.

	TERMEXT
1	hombres
2	sexo
3	género
4	clubes de sexo
5	punto de partida
6	ciudad de sao paulo
7	sexualidad
8	clubes
9	bares de sexo
10	saunas

Tabla 10. Resultados tras el análisis con TERMEXT

Esta lista de candidatos tiene varias ventajas. Todas las expresiones son sintagmas nominales, lo que no debe sorprendernos, ya que el método etiqueta el documento para considerar sólo sintagmas nominales. Además, más de la mitad son unigramas y caracterizan bien al documento. Como ya lo hemos dicho, la desventaja es que precisamente los patrones sintácticos de estos sintagmas están predefinidos.

AlchemyAPI

Según la página de AlchemyAPI, su aplicación utiliza aprendizaje de máquina y procesamiento del lenguaje natural para analizar contenido web basado en texto, para identificar: personas, organizaciones, ciudades y otra información relevante con el fin de etiquetar páginas web. Maneja los siguientes idiomas: inglés, francés, alemán, italiano, portugués, ruso, español y sueco. Por tratarse de una aplicación comercial, no se brinda más información sobre los métodos empleados por el extractor para determinar la relevancia de términos. Los mejores diez candidatos según esta aplicación son:

	AlchemyAPI T. E.
1	Sao Paulo
2	Georges Bataille
3	distintos marcadores
4	cine porno
5	Maria Filomena Gregori
6	sitios web
7	famosa avenida vieira
8	vale do anhangabaú
9	LA MEDIA LUZ
10	calle amaral gurgel

Tabla 11. Resultados tras el análisis con AlchemyAPI T.E.

Translated Labs

De acuerdo con la información contenida en la página de esta aplicación, se trata de un desarrollo de un centro de investigación donde expertos en tecnologías de la información trabajan con lingüistas para desarrollar programas con la finalidad de hacer que éstos se expresen como si fueran humanos. Actualmente sus áreas de investigación son: búsqueda semántica, traducción estadística, traducción automática, modelado del lenguaje y clasificación automática de datos. Al igual que

la aplicación anterior, no se brinda información detallada sobre la obtención de la relevancia de términos. Sus mejores diez candidatos son:

	Translated Labs T.E
1	sexo
2	sexuales
3	clubes
4	relaciones sexuales
5	aquellos cuya
6	los clubes
7	las prácticas
8	entre sexo
9	campo etnográfico
10	teniendo como

Tabla 12. Resultados tras el análisis con Translated Labs T.E.

Como en los resultados de nuestra aplicación omitimos los unigramas, para hacer una comparación. De hecho, se realizó lo mismo con los resultados de las demás aplicaciones. La tabla con los resultados de las aplicaciones sin unigramas se presenta a continuación.

	Nuestra aplicación	TERMEXT	AlchemyAPI	Labs Translated
1	sao paulo	clubes de sexo	Sao Paulo	relaciones sexuales
2	ciertos estereotipos	punto de partida	Georges Bataille	aquellos cuya
3	georges bataille	ciudad de sao paulo	distintos marcadores	los clubes
4	glory holes	marcadores de diferencia	cine porno	las prácticas
5	cine porno	bares de sexo	Maria Filomena Gregori	entre sexo
6	camas colectivas	locales comerciales	sitios web	campo etnográfico
7	locales comerciales	opciones eróticas	famosa avenida vieira	teniendo como
8	cuarto oscuro	clubes nocturnos	vale do anhangabaú	cuerpos como
9	matriz cultural	centro de sao paulo	LA MEDIA LUZ	entre el género
10	opciones eróticas	salas de cine porno	calle amaral gurgel	macrae incluye

Tabla 13. Resultados de las 4 aplicaciones sin unigramas

Notamos que los resultados brindados por el extractor de Labs Translated ofrecen menos expresiones bien formadas y esto no sorprende, ya que no cuentan con soporte para el idioma español. Por esto, y considerando que nuestro diseño de encuesta para la evaluación final es complejo, decidimos omitir los resultados de Labs Translated para el siguiente análisis. Así que la tabla definitiva se muestra a continuación:

	Nuestra aplicación	**TERMEXT**	**AlchemyAPI**
1	sao paulo	clubes de sexo	Sao Paulo
2	ciertos estereotipos	punto de partida	Georges Bataille
3	georges bataille	ciudad de sao paulo	distintos marcadores
4	glory holes	marcadores de diferencia	cine porno
5	cine porno	bares de sexo	Maria Filomena Gregori
6	camas colectivas	locales comerciales	sitios web
7	locales comerciales	opciones eróticas	famosa avenida vieira
8	cuarto oscuro	clubes nocturnos	vale do anhangabaú
9	matriz cultural	centro de sao paulo	LA MEDIA LUZ
10	opciones eróticas	salas de cine porno	calle amaral gurgel

Tabla 14. Comparación de resultados de las diferentes aplicaciones

Para evaluar la pertinencia de estas tres listas, volvimos a recurrir a un conjunto de personas que leyeron un resumen del artículo (ver anexo D) y respondieron a la pregunta: “¿Cuál lista describe mejor el contenido del artículo?”.

Cabe mencionar las dificultades a que nos enfrentamos en este nuevo experimento. Primero, el artículo era demasiado largo para ser leído rápidamente, por lo que recurrimos a resumirlo. Segundo, hubo muy poca disposición de nuestro entrevistados a leer el resumen, por lo que fueron muy pocos (sólo 8) y aparentemente escogieron las expresiones mejor formadas y no las más representativas del resumen. Además, cabe recordar que ninguno era experto en el tema del documento. Suponemos que por todas estas razones los resultados no nos favorecieron. De hecho, estuvieron abrumadoramente a favor de TERMEXT, cosa entendible, ya que considera la estructura morfosintáctica de las palabras clave. La tabla siguiente resume los resultados:

	Nuestra aplicación	TERMEXT	AlchemyApi	Total
votos	1	7	0	8

Tabla 15. Resultados de la segunda encuesta

Uso de un archivo histórico

El análisis anterior se llevó a cabo en el documento mencionado, sin hacer uso de un archivo histórico. Como se mencionó en el marco teórico, el archivo histórico es una lista que contiene los resultados de la aplicación tras analizar diversos artículos relacionados con un tema en específico. Posteriormente, se realizó el análisis con la historia resultante tras haber procesado 10 documentos referentes al tema: *sexualidad*. Los resultados obtenidos se muestran a continuación:

Sin historia	Con historia
sao paulo	sao paulo
ciertos estereotipos	ciertos estereotipos
georges bataille	georges bataille
glory holes	glory holes
estuviesen dispuestos	estuviesen dispuestos
cine porno	cine porno
camas colectivas	camas colectivas
locales comerciales	locales comerciales
sin embargo	aquellos cuya
aquellos cuya	bienestar físico

Tabla 16. Comparación de resultados: con y sin historia

En este caso la expresión *sin embargo*, que es una conjunción adversativa (y está bien formada), no sirve como palabra clave y salió de los resultados con historia, lo que ya es un avance. Afortunadamente, se remplaza por *bienestar físico*, que parece una mejor palabra clave. Por otra parte, se entiende que la cadena *aquellos cuya*, un error evidente, suba una posición (sólo ocurre en una vez en la historia de entrenamiento). Se espera entonces, que conforme vaya creciendo el número de documentos contenido en el archivo *historia*, los resultados mejoren.

Con una historia de este tipo, esperamos que la rareza de las expresiones, típica de las palabras clave, sea más evidente.

Como ya lo establecimos, para obtener resultados adecuados no nos propusimos realizar un etiquetado lingüístico previo, ni filtrado alguno. La información en la que nos basamos es la información estadística contenida en el propio texto, ya que se trató de realizar la aplicación de manera independiente al tema e idioma. Si bien los resultados al principio no parecen ser del todo adecuados, se nota que la mayoría son expresiones bien formadas, y en caso de que se incluya dentro de un tema en específico, el archivo *historia* resulta muy importante para descartar expresiones temáticamente no relevantes. Dada la independencia de la estructura del texto, la aplicación se puede adaptar prácticamente a cualquier idioma que tenga un *lematizador Porter* definido. En la aplicación final, sólo se incluyeron el español e inglés, por ser los más empleados en los documentos científicos en México.

Evaluación por un método automático

Como ya se mencionó existieron muchos inconvenientes al momento de realizar la encuesta: la disposición de las personas, el desconocimiento del idioma inglés y la falta de especialistas en el área, por mencionar los que suponemos más relevantes. Por este motivo, el resultado obtenido sirvió solamente para darnos una idea de cómo son percibidos, por parte del público no especializado, los resultados de nuestra aplicación en comparación con los de otras. Conscientes de todo esto, buscamos otra manera de llevar a cabo la evaluación sin la intervención de seres humanos.

La manera elegida fue utilizar el sistema FRESA (*Framework for Evaluating Summaries Automatically*). Este sistema fue desarrollado por el Dr. Juan Manuel Torres Moreno, quien tiene su línea de investigación en métodos de resumen automático y otras aplicaciones del procesamiento del lenguaje natural.

La idea detrás de recurrir a FRESA, es tratar los resultados de cualquier extractor de palabras clave como un pequeño resumen del documento analizado. Como la lista de resultados debe de englobar el tema o temas principales del documento en cuestión, no resulta absurda esta consideración.

FRESA toma un documento como entrada y genera un resumen de éste. Posteriormente compara el resumen que generó con otro resumen que se le proporcionó. Esta comparación se lleva a cabo mediante divergencias estadísticas que existen entre ambos documentos. Las divergencias que usa son las de: 1) Kullback-Leibler y 2) Jensen-Shannon, que son bastante conocidas y empleadas en el área (para descripciones detalladas de estas estadísticas ver Chin *et al* 2006). En pocas palabras, el resultado que generan es el valor de divergencias, que mientras menores sean, implican una mejor calidad del resumen analizado.

Análisis de los resultados

Nuevamente utilizamos los resultados de: AlchemyAPI, TERMEXT y de nuestra aplicación. Desafortunadamente la versión disponible en internet de FRESA, sólo permite textos menores a 10,000 caracteres (lo que es insuficiente para analizar el artículo empleado). Por lo tanto, se decidió utilizar como documento de entrada únicamente la lista de bigramas con al menos dos apariciones en el artículo. FRESA calculó las divergencias entre nuestros resultados y aquellos de las otras aplicaciones a evaluar y el resumen generado internamente (por Fresa) a partir de la lista de bigramas. Los resultados se muestran a continuación.

Kullback-Leibler:

TERMEXT	18.40145
Nuestra aplicación	19.13455
AlchemyAPI	19.22939

Tabla 17. Comparación de divergencia Kullback-Leibler

Jensen-Shannon:

TERMEXT	3.83014
Nuestra aplicación	3.97858
AlchemyAPI	3.98603

Tabla 18. Comparación de divergencias Jensen-Shannon

En ambas divergencias nuestra aplicación ocupa el segundo lugar, el primero, como esperábamos (por el etiquetado POS a priori), es ocupado por TERMEXT. Por otra parte, nuestra aplicación muestra una ligera ventaja en comparación con el método comercial AlchemyAPI, que probablemente también etiqueta categorías gramaticales (existen etiquetadores para todas las lenguas que manejan).

Pese al detalle del límite de caracteres que recibe el sistema FRESA, que dio origen a la sustitución del artículo por una lista de bigramas, se puede considerar como un resultado valioso ya que fue evaluado con un método automático.

Observaciones generales

En este capítulo tratamos de ordenar diversas aplicaciones, entre ellas la nuestra, de acuerdo a la calidad de los resultados que ofrecen. Esta búsqueda se llevó a cabo mediante dos aproximaciones: la primera fue realizar dos encuestas y en la segunda tratamos de reducir subjetividades mediante el uso de un método automático. En ambos casos nuestra aplicación se ubicó en la segunda posición, siendo ésta una buena ubicación ya que la aplicación no realiza ningún tipo de etiquetado.

Capítulo 7. Conclusiones y trabajo futuro

Conclusiones

Se cumplió el objetivo del presente trabajo, ya que se desarrolló el extractor de palabras clave utilizando solamente métodos estadísticos. Si bien existen resultados poco relevantes a los documentos analizados, se demostró que al hacer uso de un archivo histórico del tema tratado por el documento, éstos tienden a reducirse para dar paso a los n-gramas más relevantes.

Otro modo de la aplicación que mejora considerablemente la calidad de los resultados, es el que hace uso de una lista de PC validada por expertos en un tema específico. Desgraciadamente no siempre se puede contar con esta lista, de modo que en la presentación de los resultados de este trabajo este modo se omitió.

Trabajo futuro

Soporte para más idiomas: la aplicación actualmente sólo es capaz de analizar adecuadamente dos idiomas: el español y el inglés. Si bien la aplicación es capaz de analizar texto escrito en cualquier idioma (exceptuando idiomas como el chino, japonés, coreano, entre otros; que hacen uso de ideogramas), los resultados no serían tan buenos ya que no utiliza un lematizador adecuado a estos idiomas. Debido a la naturaleza de los documentos científicos sólo se optó por el inglés, pero se podrían incluir otros idiomas como el: francés, italiano, alemán, holandés, finés, portugués, entre otros.

Agregar un tercer modo de funcionamiento: en este modo el sistema será capaz de corroborar la consistencia de las PC propuestas por el usuario, asignándole una calificación de pertinencia a cada una. Tentativamente este proceso se realizaría mediante la comparación de las dos listas: las PC del usuario y las generadas por la aplicación. Desgraciadamente este modo requiere el análisis de especialistas en cada tema, por lo que la implementación de esta fase fue inviable en

el desarrollo del presente trabajo. Pero contando con un periodo más grande de tiempo y con la ayuda de especialistas de los diversos temas se podría implementar.

Investigación de la incorporación de un etiquetador automático: investigar la existencia de algún etiquetador automático de categorías gramaticales que funcione independientemente del idioma. En su defecto, desarrollar uno, siendo conscientes que su desarrollo e implementación podría tardar varios años de investigación. Aunado a esto vendría la necesidad de determinar automáticamente los patrones de categorías gramaticales de las palabras clave y términos de las áreas de especialidad, para no tener que declararlas a priori.

Anexos

Anexo A: contiene una breve descripción del perceptrón utilizado para el cálculo del Índice de Gramaticalidad

Anexo B: contiene la descripción del algoritmo de Porter en las dos variantes que se utilizaron en el desarrollo de la aplicación: para el idioma inglés y la adaptación al idioma español.

Anexo C: muestra la encuesta que se realizó para elegir la combinación de resultados que ayudara más a la deducción de un tema.

Anexo D: muestra la encuesta donde se realizó la comparación de resultados de nuestra aplicación con otras dos.

Anexo A. Perceptrón

El Perceptrón es una red neuronal formada por una sola neurona capaz de clasificar vectores de entrada en dos categorías. Cada vector de entrada representa a un elemento que se desea clasificar y los valores del vector son propiedades de dicho elemento que son necesarias para diferenciar a dicho elemento e incluirlo en cualquiera de las categorías.

De forma básica, el perceptrón se entrena para encontrar la solución más óptima representada por una recta o plano denominado límite de decisión, que se representa por una ecuación cuyo orden es igual al de los vectores de entrada.

Para este proyecto, el perceptrón proporciona una herramienta necesaria para generar un índice capaz de medir qué tanto una palabra gráfica pertenece al grupo de palabras función; tal índice, entre cero y uno, en este trabajo es denominado *índice de gramaticalidad*, a continuación se muestra de forma gráfica el perceptrón usado en este proyecto:

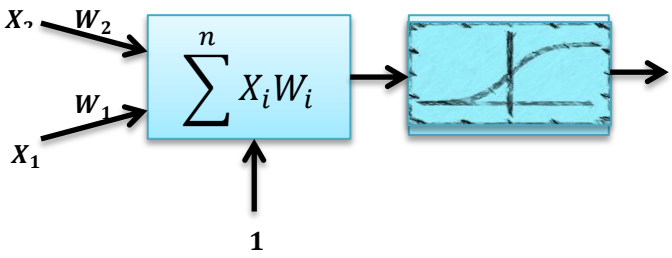

Figura 16. Perceptrón utilizado en el proyecto

El perceptrón anterior se compone de dos procesos, el primero consta en la suma de los productos de los elementos de dos vectores: el vector de entrada X y el vector de pesos W, el segundo se trata de la función de transferencia de este vector.

La imagen anterior es una representación genérica del perceptrón usado en el proyecto, donde, en este caso X es el vector de entrada que representa a cada palabra y X1 es la frecuencia de la palabra en los documentos de entrenamiento y X2 es la longitud de la palabra, en la tercer entrada como se trata del término in-

dependiente siempre será 1 y se multiplicará por el sesgo B. La función de transferencia usada es la función logsig que se interpreta de la siguiente manera:

$$logsig(n) = \frac{1}{1 + e^{-n}}$$

El objetivo de este perceptrón es el de separar las palabras función de las palabras contenido, debido a la naturaleza misma del lenguaje es imposible generar un límite de decisión cien por ciento eficaz, por lo que se decidió hacer uso de la función de transferencia logsig para mostrar un coeficiente entre cero y uno en lugar de un valor determinista de cero o uno.

El entrenamiento de la neurona se lleva a cabo con documentos representativos del idioma, tanto inglés como español y consiste en comparar las palabras de estos documentos contra las palabras de una lista de paro que contiene un conjunto de palabras función que servirán como valores de entrenamiento al perceptron, el algoritmo se explica a continuación.

Algoritmo de entrenamiento

```
Entrenamiento()
Para(j=0;hasta j=total_iteraciones )
Para(i=0;hasta i=longitud de lista)
        P = Obtener_elemento( i )
    fr = P.frecuencia( );
    l = P.longitud( );
    R = función( fr , l );
    Si (P es funcional & R<0.5) Entonces
        W1+=(1-R)*alpha*fr;
        W2+=(1-R)*alpha*l;
        B+=(1-R)*alpha;
    Sino Si(P no es funcional & R>0.5) Entonces
        W1+=(-R)*alpha*fr;
        W2+=(-R)*alpha*l;
        B+=(-R)*alpha;
    Fin Si
R_viejo=R;
Fin Para
Si((R_viejo-R)<delta)
    Romper_Para
Fin Para

función( freq , long )
    n = W1*freq + W2*long + B
    r = 1/(1-e^-n)
Regresa r
```

Donde alpha es la tasa de aprendizaje que se proporciona al inicio y W1 W2 y B son los pesos de la neurona con valores iniciales:

$$\boldsymbol{W1 = 0.1}$$

$$\boldsymbol{W2 = 0.1}$$

$$\boldsymbol{B = 0.0}$$

El método de Aprendizaje se basa en el aprendizaje de Hebb donde se obtiene la diferencia entre el valor esperado y el valor obtenido y se multiplica por el producto de la entrada y la tasa de aprendizaje:

$$W_1 = W_1 + \Delta W_1$$

$$\Delta W_1 = (V_e - V_r) * \alpha * x$$

El aprendizaje se realiza cuando el perceptrón clasifica erróneamente cualquiera de las palabras, es decir, cuando asigna un valor mayor a 0.5 a una palabra que no aparece en la lista de paro o cuando asigna un valor menor a 0.5 para palabras que sí aparezcan en la lista de paro.

El proceso se repite hasta que se alcanza el número máximo de iteraciones o hasta que se llegue a un valor de convergencia, es decir, cuando la diferencia entre los resultados de la neurona no se modifique o se modifique muy poco entre cada iteración. Esto se logra al realizar una comparación en cada iteración entre el valor nuevo y el valor viejo y comprobar que la diferencia no sea mayor que un valor delta preestablecido para el entrenamiento.

Anexo B. Algoritmo de Porter

A continuación se presenta el algoritmo empleado en el presente trabajo en sus dos variantes: la versión para el idioma inglés y la adaptación al idioma español. Como se mencionó en el capítulo 4 la versión del algoritmo de Porter que utilizamos es la implementación en Java del proyecto escrito en el lenguaje de programación *Snowball*[8].

a) Idioma inglés

Definiciones

- Se considera *"vocal"* a una de las siguientes letras: *y, a, e, i, o,* y *u*
- Se considera *"doble"* a uno de los siguientes pares de letras: *bb, dd, ff, gg, mm, nn, pp, rr,* y *tt*
- Una "*terminación li válida"* es*: c, d, e, g, h, k, m, n, r,* y *t*
- *R1*: es la región después de la primera "*no-vocal"* seguida de una "*vocal"*, o el final de una palabra si no existe dicha "*no-vocal"*.
- *R2*: es la región después de la primera "*no-vocal"* seguida de una *vocal* en *R1*, o el final de una palabra si no existe dicha no vocal.
- En una palabra una "*sílaba corta"* es: una "*vocal"* seguida por una "*no-vocal"* diferente de *w, x* o *y* y precedida por una "*no-vocal"* o una "*vocal"* seguida por una "*no-vocal"* al principio de la palabra. De esta manera *rap*, *trap*, *entrap* terminan con una "*sílaba corta"*, y *ow*, *on* y *at* son tipos de "*sílabas cortas"*. Por otro lado, *uproot*, *bestow*, *disturb* no terminan con una "*sílaba corta"*.
- Una palabra es denominada "*palabra corta"* si termina con una "*sílaba corta"* y si *R1* no contiene letras. Así: *bed*, *shed* y *shred* son "*palabras cortas"*, sin embargo, *bead, embed* y *beads* no son "*palabras cortas"*.

[8] Adaptaciones del algoritmo de Porter en diferentes idiomas : http://snowball.tartarus.org/algorithms/

Consideraciones

- Un "*apóstrofe"* (') es considerado como una letra.
- Si una palabra contiene solamente dos letras o menos se deja la palabra intacta, de otro modo se realizan las siguientes acciones:
    - Remueve "*apóstrofes"* iniciales en caso de existir, después cambia las *y* existentes a *Y* y establece las regiones *R1* y *R2*.
- Después se realizan los siguientes pasos.

Algoritmo

0. Busca: *', 's* o *'s'* al final de la palabra y en caso de encontrarlos se eliminan.

1.

a) Busca los siguientes sufijos y en caso de encontrarlos realiza la acción indicada:
sses: reemplaza por *ss*
ied+ *ies*+: reemplaza por *i* si es precedido por más de una letra, en otro caso por *ie*.
s: elimina en caso de que la letra anterior no sea una "*vocal"*
uss+, *ss*: no se hace cambio alguno

b) Busca los siguientes sufijos y en caso de encontrarlos realiza la acción indicada

eed, *eedly*+: si está en *R1* reemplaza por *ee*.

ed, *edly*+, *ing*, *ingly*+: elimina el sufijo si la letra anterior es una "*vocal"*, después de eliminarlo:

si la palabra termina con: *at*, *bl* o *iz*: añade *e*

si la palabra termina con “*doble”*: elimina la última letra

si termina con una "*palabra corta”* añade *e*

c) Reemplaza el sufijo *y* o *Y* por *i* si está precedida por una "*no vocal”* y no es la primer letra de la palabra.

2. Busca los siguientes sufijos y si se encuentran en *R1* realiza lo siguiente:

enci reemplazar por *ence*

tional: reemplazar por *tion*

anci: reemplazar por *ance*

abli: reemplazar por *able*

entli: reemplazar por *ent*

izer o *ization*: reemplazar por *ize*

ational, ation o *ator*: reemplazar por *ate*

alism, aliti o *allí*: reemplazar por *al*

fulness: reemplazar por *ful*

ousli o *ousness*: reemplazar por *ous*

iveness, iviti: reemplazar por *ive*

biliti o *bli*+: reemplazar por *ble*

ogi+: reemplazar por *og* si está precedido por *l*

fulli+: reemplazar por *ful*

lessli+: reemplazar por *less*

li+: elimina si es precedido por una "*terminación li válida"*.

3. Busca los siguientes sufijos y si se encuentran en *R1* realiza lo indicado

tional+: reemplaza por *tion*

ational+: reemplaza por *ate*

alize: reemplaza por *al*

icate, iciti o *ical*: reemplaza por *ic*

ful o *ness*: elimínalos

*ative**: si se encuentra en *R2* elimínalo

4. Busca los siguientes sufijos y si se encuentran en *R2* realiza lo indicado:

al, ance, ence, er, ic, able, ible, ant, ement, ment, ent, ism, ate, iti, ous, ive o *ize*: elimínalo
ion: si está precedido por *s* o *t*, elimínalo

5. Busca los siguientes sufijos y si se encuentran realiza lo indicado

e: se elimina si se encuentra en *R2* o si está en *R1* y no es precedido por una "*sílaba corta"*.
l: se elimina si se encuentra en *R2* y está precedido por *l*

Finalmente en caso de existir *Y* se cambian a *y*.

b) Idioma español

Definiciones

- Se añaden las siguientes letras: *á, é, í, ó, ú, ü* y *ñ*
- Se considera "*vocal"* a: *a, e, i, o, u, á, é, í, ó* y *ú*
- Se considera "*consonante"* cualquier letra que no sea "*vocal"*.
- *R2* se define igual que en el algoritmo para el idioma inglés
- Si la segunda letra es una consonante, *RV* es la región después de la siguiente vocal, en el caso de que las dos primeras letras sean vocales, *RV* es la región después de la siguiente *"consonante"*

Algoritmo

Siempre realizar el paso 0, el paso 1 y el paso 3.

0. Pronombres:

   Busca el sufijo más largo de la siguiente lista: *me, se, sela, selo, selas, selos, la, le, lo, las, les, los, nos*, elimínalo en caso de que esté en *RV* y se encuentre después de:

   a*) iéndo, ándo, ár, ér, ír*

   b) *ando, iendo, ar, er, ir*

   c) *yendo* siguiendo una *u*

   En el caso c) la *u* puede estar fuera de *RV*. En el caso a) después de la eliminación se quita la acentuación, por ejemplo: *haciéndola* quedaría como *haciendo*.

1. Eliminación de sufijos

   Busca el sufijo más largo de las siguientes listas y si se encuentran en R2, realiza lo indicado:

anza,anzas, ico, ica, icos, icas, ismo, ismos, able, ables, ible, ibles, ista, istas, oso, osa, osos, osas, amiento, amientos, imiento, imientos: elimínalo

adoraa, ador, ación, adoras, adores, aciones, ante, antes, ancia, ancias: elimínalo, si es precedida por *ic* y si éste se encuentra también en *R2*, elimínalo.

logia, logías: reemplaza con *log*

ución, uciones: reemplaza con *u*

encia, encías: reemplaza con *ente*

mente: elimínalo, si es precedido por *ante, able* o *ible* y éste se encuentra también en *R2* elimínalo de igual forma.

idad, ida: elimínalo, si es precedido por: *abil*, *ic* o *iv* y éste se encuentra también en *R2* elimínalo de igual forma.

iva, ivo, ivas, ivos: elimínalo, si es precedido por: *at* y éste se encuentra también en *R2* elimínalo de igual forma.

Si el sufijo *amente* se encuentra en *R1* es eliminado, si es precedido por *iv* y éste se encuentra en *R2* se elimina. Si a su vez *iv* es precedido por *at*, *os*, *ic* o *ad* y se encuentra en *R2*, el sufijo es eliminado.

2. Si no se eliminó ningún sufijo en el punto 1, busca el sufijo más largo de la siguiente lista que esté en *RV*: *ya, ye, yan, yen, yeron, yendo, yo, yó, yas, yes, yais, yamos*: elimínalo si es precedido por *u*.

   En caso de que no se haya eliminado el sufijo en el paso previo, busca el sufijo en *RV* de las siguientes listas y realiza lo indicado:

   en, es, éis, emos: elimínalo y si es precedido por *gu* elimina la *u*.

arían, arías, arán, arás, aríais, aría, aréis, aríamos, aremos, ará, aré, erían, erías, erán, erás, eríais, ería, eréis, eríamos, eremos, erá, eré, irían, irías, irán, irás, iríais, iría, iréis, iríamos, iremos, irá, iré, aba, ada, ida, ía, ara, iera, ad, ed, id, ase, iese, aste, iste, an, aban, ían, aran, ieran, asen, iesen, aron, ieron, ado, ido, ando, iendo, ió, ar, er, ir, as, abas, adas, idas, ías, aras, ieras, ases, ieses, ís, áis, abais, íais, arais, ierais, aseis, ieseis, asteis, isteis, ados, idos, amos, ábamos, íamos, imos, áramos, iéramos, iésemos, ásemos: elimínalo.

3. Busca el sufijo de la siguientes listas y realiza lo indicado:

   os, a, o, á, í, ó: elimínalo en caso de que esté en *RV*

   e, é: elimínalos si se están en *RV*, si son precedidos por *gu* y la *u* está en *RV*, elimina la *u*.

4. Finalmente elimina los acentos agudos

Anexo C. Encuesta

Muchas gracias por su participación en este experimento.

Edad:	
Sexo:	
Educación:	

Instrucciones

1. En cada columna de cada tabla siguiente se representan diferentes posibilidades descriptivas del tema de un documento (un tema por tabla). Por favor, tacha las celdas que contienen expresiones que NO te ayuden a predecir el tema del documento.

   IMPORTANTE: para analizar cada celda hay que considerar si la expresión que contiene está completa y si se **refiere a un único concepto:**

Ejemplos	Expresión	Concepto	Por lo tanto
Lo	Incompleta (es sólo un artículo)	No identificable	**X**
Género	Completa (es un sustantivo)	Identificable	
en la	Incompleta (no hay sustantivo ni verbo)	No identificable	**X**
migración internacional	Completa (dos palabras que funcionan como sustantivo)	Identificable	
las relaciones son	Le sobra material (es sustantivo y verbo)	Se designan dos conceptos, el de relación y el de ser	**X**
Rol de género	Completa (tres palabras que funcionan como sustantivo)	Identificable	

¿De qué tema crees que se trata el documento 1?

¿De qué tema crees que se trata el documento 2?

Documento 1

Sin TF-IP	C-value	IG	AM
el género y la	de la	antropologí?a impropia	sao paulo
sao paulo	en la	camilo albuquerque	de la
los clubes y bares	de los	universidad estatal	cine porno
georges bataille	en el	diseño sustancializa	ciertos estereotipos
sin embargo	de las	aportar pruebas	georges bataille
camas colectivas	en los	pruebas empíricas	glory holes
en la producción	el género y la sexualidad	maria filomena	en la
así como	y la	violencia interpersonal	locales comerciales
cine porno	a la	mutuamente imbricadas	ya sea
aquellos cuya	y bares de sexo	occidentales contemporáneas	sin embargo
matriz cultural	el género	entidades biológicas	en el
y simíµes	la sexualidad	biológicas preexistentes	camas colectivas
clubes nocturnos	como un	espíritu contestatario	estuviesen dispuestos
estas páginas	parte de	gender trouble	de los
cabinas privadas	en las	post- estructuralista	entre hombres

LL	Sin IG	Sin C-value	Todas las medidas
sao paulo algunos clubes presunta-	de la	sao paulo	el género y la
sao paulo por lo menos	de los	ciertos estereotipos	sao paulo
sao paulo contemporáneo	en la	georges bataille	los clubes y bares
sao paulo algunos clubes	en el	glory holes	sin embargo
sao paulo algunos	sao paulo	estuviesen dispuestos	así como
sao paulo por lo	en los	cine porno	en la producción
sao paulo por	de las	camas colectivas	georges bataille
sao paulo	el género	locales comerciales	y simíµes
de la avenida ipiranga macrae	lo que	sin embargo	camas colectivas
de la violencia interpersonal	la sexualidad	aquellos cuya	cine porno
de la escasa iluminación	así como	cuarto oscuro	aquellos cuya
de la labor pionera	el género y la sexualidad	matriz cultural	matriz cultural
de la violencia interpersonal y	de sexo	ya sea	punto de
de la escasa iluminación y	de la sexualidad	opciones eróticas	reflexión sobre
de la labor pionera de	sexual y	sitio web	tales como

Documento 2

AM	Sin TF-IPF	IG	Sin LL
activated sludge	tio2 nps on biological	terrestrial dry-weather	tio2 nps
publication date web	tio2 nps	debbie lee	new york
publication date	new york	tracking markers	gi illness
united states	activated sludge	dry weather	activated sludge
tio2 nps	rhode island	tank densities	key laboratory
wastewater treatment	gi illness	previously reported	rhode island
key laboratory	phosphorus removal	monte carlo	liquid culture
source waters	land-based runoff	incorporate uncertainties	mg/l tio2
new york	robert h	swimmer ingestion	phosphorus removal
gi illness	forward osmosis	ingestion volumes	fire fighting
rhode island	fo membrane	dose–response parameters	publication date
phosphorus removal	manganese peroxidase	magnitude greater	spring creek
mg/l tio2	crediting system	clostridium perfringens	bacterial community
mg/l tio2 nps	log baffish	important assumptions	titanium dioxide
liquid culture	ghg crediting	pathogens coming	denaturing gradient

C-value	Sin IG	LL	Todas las medidas
in the	tio2 nps	activated sludge microbial communities	tio2 nps
of the	activated sludge	activated sludge microbial communities is	tio2 nps on biological
publication date web	publication date web	activated sludge microbial communities in	activated sludge
united states environ	united states environ	activated sludge microbial communities	new york
tio2 nps	source waters	activated sludge xiong zheng	rhode island
source waters	united states	united states ch2m hill	gi illness
activated sludge	in the	activated sludge xiong	phosphorus removal
wastewater treat-	wastewater treatment	united states §school of engineering	united states
on the	mg/l tio2 nps	united states environ	source waters
university of	of the	united states §school	adjacent to
source of	new york	united states ch2m	wastewater treat-
mg/l tio2 nps	phosphorus removal	united states §school of	liquid culture
key laboratory of	key laboratory of	activated sludge microbial	nankai university
in this	tio2 nps on biological nitro-	activated sludge are sparse	water flux
due to the	wastewater treatment	united states department of earth	land-based runoff

Anexo D. Comparación de resultados

Instrucciones

Gracias por su participación en este experimento, a continuación se presenta un extracto del siguiente artículo:

Título: Machos a la media luz: miradas de una antropología impropia
Autor: Camilo Albuquerque de Braz

Después de leerlo por favor elija una lista, de las tres disponibles en la última página, que mejor describa al texto.

Machos a la media luz: miradas de una antropología impropia

Resumen

Este artículo analiza cómo operan distintos marcadores de diferencia en clubes y bares de sexo para encuentros sexuales entre hombres (ubicados en locales comerciales en la ciudad de Sao Paulo, Brasil) y en la producción de sus cuerpos como deseables y sujetos inteligibles. Además reflexiona sobre opciones eróticas y prácticas sexuales, que exige el cuestionamiento sobre *erotismo* expresado como concepto por teóricos como Georges Bataille.

Introducción

La metodología usada en esta investigación pretende ser antropológica, ya que trata de interpretar un sinnúmero de discursos, charlas, experiencias; partiendo de la traducción para un lenguaje técnico. Continúa la línea de estudios iniciada por María Filomena Gregori, quien ha profundizado en el campo de la Antropología y Estudios de Género sobre las formas de erotismo contemporáneo y con temas de la violencia interpersonal y de género.

Diferencias

Un problema en la producción académica de temas de sexualidad es que en sociedades occidentales contemporáneas adquieren diferentes significados para los sujetos que integran los diferentes segmentos sociales. Esto justifica la importancia de los estudios comparativos, los interesados en saber el significado dado por el sexo y la reproducción en diferentes contextos culturales y sociales.

En Thinking Sex (Rubin, 1993), Gayle Rubin trabaja desde la perspectiva "foucaultiana"• para proponer elementos conceptuales y descriptivos para reflexionar sobre el género y la política. En la década de los 90, se vio la gran cantidad de "estudios gays y lésbicos", exigiendo una distinción entre el género y sexualidad, trazando el mapa de la "estratificación sexual" presente en las sociedades modernas.

Una crítica de estos estudios es que en los análisis de las sexualidades heterosexuales, las cuestiones de género parecen atrapadas en una distinción binaria, entre el "sexo" y el "género". La noción de que las prácticas sexuales son "buenas" y "malas" impregna buena parte de esta generación, a pensar sobre el sexo como un vector de opresión ejecutado a través de otras formas de desigualdad social (clase, raza, origen étnico o de género).

En Gender Trouble, Judith Butler facilita la convergencia entre las perspectivas feministas, gays y lesbianas de género con la teoría post-estructuralista. La crítica del sujeto no es una negación o rechazo de ello, sino una forma de cuestionamiento sobre su construcción. Se pueden apuntar las ideas de Avtar Brah para pensar en la identidad como la posicionalidad, para hacer la separación entre "teoría de la sexualidad" y "teorías de género", centrándose en la primera y dejando la segunda para el feminismo.

Contextos

La región del centro de Sao Paulo, es un lugar frecuentado por hombres que mantienen relaciones afectivo-sexuales con otros hombres. En la década de 1960 se

abren en Sao Paulo clubes destinados a un cliente "homosexual" de clase media, "que buscaba sitios de encuentros donde hubiese más seguridad contra ataques de la policía y de agresores". Después de la apertura política, aumenta el número de establecimientos del denominado "mercado gay".

Sus tendencias se agruparon no sólo por la orientación sexual, sino por sexo, el consumo de energía, el "estilo", por la forma a partir de la cual expresan sus preferencias sexuales. Franí§a resalta la importancia de la difusión de imágenes, de estilos, hábitos y actitudes relacionados con la política de las identidades y las culturas emergentes a la identidad gay. Este movimiento incluye Internet, donde surge la sigla GLS (gays, lesbianas y simpatizantes), propagada por el MixBrasil, de 1994, que incluía un sitio Web y un festival de cine y cultura "alternativa", hacia ese nuevo público.

En el "mercado del sexo" o "mercado contemporáneo de bienes eróticos" en Sao Paulo, existe un amplio y diverso segmento de personas que buscan relaciones sexuales con otras del mismo "sexo", incluidos los hombres que buscan otros hombres. Hay muchos clubes nocturnos y bares con un espacio específico para los encuentros sexuales. Lo que más me llamó la atención en dichos sitios era que en la gran mayoría de los perfiles registrados los usuarios buscaban hombres, "con actitud masculina", machos, "sin plumas.

La "hiper-valoración de la masculinidad" y la producción de "machos" como sujeto y objeto de deseo son elementos que intervienen en los procesos de materialización de los cuerpos y en la producción de subjetividades en muchos contextos de tránsito de hombres que se relacionan afectivo-sexualmente con otros hombres. Por más cuestionable en términos de jerarquías que se sitúe, la creación discursiva de "machos" como sujetos de deseo entre esos hombres se puede leer, quizás, como rearticulación o como el desplazamiento de las convenciones relativas al sexo, género, deseo sexual y las prácticas sexuales y corporales que componen la matriz cultural heteronormativa por medio de la cual se gana inteligibilidad, o sea, "se existe".

¿Cómo se da combinación de estos elementos en los comercios para encuentros sexuales de hombres o, más específicamente, en los clubes de sexo? y ¿a partir de cuáles etiquetas de los cuerpos de los sujetos se materializan esos lugares, sea como deseables, sea como rechazables?

Formulé estas y otras preguntas similares, pero seguía sin resolver el problema de definir el alcance de las investigaciones. Los clubes o bares de sexo para hombres son un fenómeno transnacional, con sus homólogos en las "escenas" gays de América del Norte y Europa. El primer sitio de comercio sexual entre hombres que se distingue del "modelo" adoptado por la sauna fue el Station, un cruising-bar que abrió sus puertas en 1998 en el barrio Pinheiros. Es común entre los propietarios de los clubes de sexo la afirmación de que Station abrió el camino para la aparición de muchos de ellos en la ciudad.

Ese club cerró aproximadamente dos años después de abierto y fue reabierto en otra área cercana, en el Largo do Arouche, manteniendo el nombre, equipos, accesorios y el título de "primer sex club en Brasil", como se muestra tanto en su sitio de Internet como en los volantes que revelan su programación.

Yendo hacia los barrios de "clase media-alta", están, además del mencionado Station, los otros dos clubes que investigué. El Gladiators surgió hace más de 4 años y se encuentra al lado del centro comercial Frei Caneca, en el barrio Consolaí§ao.

El club RG ha surgido auto-afirmándose como un club privado, "no abierto al público en general"• . Inaugurado poco después del Blackout en el barrio de Marqués de Sade y de Leopold von Sacher-Masoch.

A partir de aquí, traigo más datos de campo etnográfico fusionados con el análisis de las entrevistas con algunos de los frecuentadores y empresarios de esos clubes, con el fin de reflexionar sobre cómo distintos marcadores de diferencia operan en esos lugares, de manera interseccionada, a fin de producir sus sujetos inteligibles y cuerpos (in)deseables.

Metodologías

Cuando empecé a incursionar en campo, lo hice bajo el espectro de los riesgos éticos que podría implicar la etnografía. Algunos no asumen o "visibilizan" fuera de los LCES sus preferencias eróticas, sexuales. Otros son comprometidos, ya sea con las mujeres, ya sea con otros hombres. En el segundo caso, están aquellos cuya relación es "abierta", permitiendo las relaciones sexuales con otras personas. Sin embargo, existen aquellos cuya relación es "cerrada", esos hombres valorizan la "discreción"• y buscan a otros que, como ellos, garantizan su "discreción" y su "secreto"

He creado perfiles en estas páginas. En los perfiles, explicaba el tema de la investigación, ofrecía el plan de estudios y mi dirección de correo electrónico, que también utilicé como un MSN-messenger, creado específicamente para la investigación.

Especificaba que buscaba mayores de 18 años, y que mi único criterio era que hubiesen ido a locales comerciales para sexo entre varones en la ciudad de Sao Paulo por lo menos una vez. También activé una red de amigos/as, compañeros y conocidos/as que, de alguna manera, yo sabía que podrían presentarme posibles contribuyentes a la investigación.

A pesar del gran número de personas que me agregaban en MSN y, a continuación, mostraban no haber ido a ninguno de esos sitios, en poco más de dos años entrevisté por MSN a 29 hombres. Algunos de ellos nunca habían ido a un club de sexo, pero trajeron muy enriquecedoras colaboraciones sobre otros lugares. También entrevisté con grabadora a 17 frecuentadores de clubes y bares de sexo. Por otra parte, entrevisté a los empresarios y/o encargados de los clubes, para saber sobre su historia y el día a día del local desde su perspectiva. Examinado acerca de sus experiencias sexuales en diferentes contextos, especialmente en los clubes y bares de sexo, mi búsqueda fue de informes sobre sus dinámicas, sus sujetos, sus prácticas, significados, jerarquías y convenciones.

Listas de palabras

Lista 1	Lista 2	Lista 3
sao paulo	clubes de sexo	Sao Paulo
ciertos estereotipos	punto de partida	Georges Bataille
georges bataille	ciudad de sao paulo	distintos marcadores
glory holes	marcadores de diferencia	cine porno
estuviesen dispuestos	bares de sexo	Maria Filomena Gregori
cine porno	locales comerciales	sitios web
camas colectivas	opciones eróticas	famosa avenida vieira
locales comerciales	clubes nocturnos	vale do anhangabaú
aquellos cuya	centro de sao paulo	LA MEDIA LUZ
bienestar físico	salas de cine porno	calle amaral gurgel

Glosario

C/NC-value. Método que combina información lingüística y estadística para la extracción de términos multipalabra. El C-value mejora la medida estadística común de la frecuencia de ocurrencia para la extracción de términos, haciéndolo sensible a términos anidados. El NC-value proporciona un método para la extracción de palabras gráficas de los términos en su contexto e incorpora esa información en la extracción de dichos términos. Véase Frantzi (2000).

Etiquetado de categorías gramaticales (*Part-of-speech tagging*). Proceso en el cual se le asigna a cada una de las palabras en un texto su categoría gramatical.

Expresión multipalabra. Conjunto de palabras gráficas extraídas de un texto que pueden o no ser candidatas a término.

Índice de gramaticalidad. Coeficiente calculado para cada palabra en un texto que indica la probabilidad de que dicha palabra se comporte como una palabra funcional dentro del texto, basándose en su frecuencia y longitud.

Lema (*Stem*). Es la palabra que representa a todas las formas flexionadas de la misma (plural, femenino, conjugada).

Lematización (*Stemming*). Proceso lingüístico que permite obtener el lema de las palabras en un documento. Véase Porter (1980).

Lista de paro (*Stop list*). Lista de palabras que se utilizan como filtro en el análisis de textos. En este proyecto se refiere a una lista de palabras funcionales que se utiliza para el cálculo del índice de gramaticalidad.

Minería de textos (*Text Mining*). Conjunto de métodos y técnicas diseñadas para la extracción de información a partir de datos estructurados o no estructurados dentro de un conjunto de textos. Véase Kao (2007).

N-grama: Unidad lingüística formada por una o más palabras gráficas que representa un posible término o palabra clave dentro del texto del que se obtiene.

Palabra clave. Término de una o más palabras que, de acuerdo a su uso dentro de un texto, se considera representativo del contenido del texto, es decir, indica un tema o tópico tratado en el documento.

Palabra de contenido (*Content word*). Palabras que llevan el contenido o significado de la expresión donde se encuentran, también llamadas palabras léxicas incluyen verbos, sustantivos, adjetivos y adverbios.

Palabra función o funcional (*Function word*). También llamadas palabras gramaticales o vacías, son aquellas palabras en un texto que contienen menos información sobre el texto y más sobre la estructuración del discurso; esto es, sirven en general para indicar relaciones entre las palabras de contenido. Entre estas palabras se incluyen los artículos, las conjunciones, preposiciones, verbos auxiliares, etc.

Palabra gráfica. Es aquella estructura en un texto que representa una unidad lingüística, separada de otras palabras por espacios o signos de puntuación.

Término. Conjunto de una o más palabras gráficas que forman una expresión que, de acuerdo a su estructura, se considera como una expresión que representa un concepto dentro de un texto.

Terminidad (*Termhood*). Medida que indica la capacidad de una expresión para ser considerada como un término dentro de un texto. Vease Frantzi (2000).

Referencias

Albuquerque de Braz Camilo. 2009. "Machos a la media luz: miradas de una antropología impropia" en *AIBR. Revista de Antropología Iberoamericana*.

Ananiadou Sophia, McNaught John. 2006. *Text Mining for Biology and Biomedicine*. Manchester: Artech House.

Chin-Yew Lin, Guihong Cao, Jianfeng Gao, Jian-Yun Nie. 2006. “An information-theoretic approach to automatic evaluation of summaries” en *HLT-NAACL '06 Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*

Dunning Ted. 1993. "Accurate Methods for the Statistics of Surprise and Coincidence" en *Journal Computational Linguistics - Special issue on using large corpora*, volumen 19 marzo 1993.

Frantzi Katerina, Ananiadou Sophia, Mima Hideki. 2000. "Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method" en *International Journal on Digital Libraries*

Gosling James, Joy Bill, Steele Jr Guy L, Bracha Gilad, Buckley Alex. 2010. *The Java Language Specification.* Oracle America.

Göker Ayse, Davies John. 2009. *Information Retrieval: Searching in the 21st Century*. Reino Unido: John Wiley and Sons, Ltd

Hagan Martin T., Demuth Howard B, Beale Mark H. 1996. *Neural Network Design.* Pws Publisher

Han Jaiwei,Kamber Micheline, Pei Jian. 2011. *Data Mining. Concepts and Techniques*. Morgan Kaufmann.

Hulth Anette. 2003. "Enhancing Linguistically Oriented Automatic Keyword Extraction" en Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004

Jurafsky Daniel;.Martin James H. 2009. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Prentice Hall, Segunda edición.

Kageura Kyo. 1999. “Bigram Statistics Revisited: A Comparative Examination of Some Statistical Measures in Morphological Analysis of Japanese Kanji Sequences” en *Journal of Quantitative Linguistics*, volumen 6 número 2 1999

Kao Anne, Poteet Stephen R. 2007. *Natural language processing and text mining.* Londres: Springer Science

Khan Aurangzeb, Bahurdin Baharum B., Khan Khairullah. 2009. “An Overview of E-Documents Classification”

Lindholm Tim, Yellin Frank, Bracha Gilad, Buckley Alex. 2011. *The Java Virtual Machine Specification*. Oracle America.

Matsuo Y., Ishizuka M. 2004. "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information" en *International Journal on Artificial Intelligence Tools*

Mihalcea Rada. 2004. "Graph-based Ranking Algorithms for Sentence Extraction,Applied to Text Summarization" en Proceedings of Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)

Moens Marie-Francine. 2006. Information extraction: algorithms and prospects in a retrieval context . Dordrecht Springer Science

Nakagawa Hiroshi, Mori Tatsunori. 2002. "A Simple but Powerful Automatic Term Extraction Method" en *COMPUTERM 2002 — Proceedings of the 2nd International Workshop on Computational Terminology. Taipel*, Taiwan, 2002, pp. 29–35.

Poo Danny, Kiong Derek, Ashoj Swarnalatha. 2007 .*Object Oriented Programming and Java*. Springer Science.

Porter Martin F. 1980. “An algorithm for suffix stripping” en: *New models in probabilistic information retrieval*. Londres: British Library

Shannon Claude Elwood. 1948. "A Mathematical Theory of Communication" en *The Bell System Technical Journal*

Sidorov Grigori, Gelbukh Alexander, Lavin-Villa Eduardo, Chanona Liliana. 2010. “Automatic Term Extraction using Log-likelihood based Comparison with General Reference Corpus” en *Natural Language Processing and Information Systems*

Spärck Jones Karen. 1972. "A statistical interpretation of term specificity and its application in retrieval" en *Journal of Documentation* volumen 60 número 5 2004 pp. 493-502.

TERMEXT. Extractor Terminológico. Instituto de Ingeniería UNAM – Ingeniería Lingüística. http://www.iling.unam.mx/termext/

Turney Peter. 1999. "Learning to Extract Keyphrases from Text", en *National Research Council Canada*, ERB-1057, NRC-41622.

Weiss Sholom M, Indurkhya Nitin, Zhang Tong, Damerau Fred J. 2005. *Text Mining, Predictive Methods for Analyzing Unstructured Information*. Nueva York: Springer Science

Whitley Darrell. 1989. "The GENITOR algorithm and selection pressure: Why. rank-based allocation of reproductive trials is best" en *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*

Wu C. Tomas. 2010. An Introduction to Object Oriented Programming with Java. McGraw-Hill Science.