Algoritmo de Porter
Entrega Final

Cristian Arias Menares, Fabian Tapia Jorquera & Maximiliano Campos
Julio 2019

Universidad Tecnológica Metropolitana (UTEM).
Departamento de Informática.
Teoría de Autómatas y Lenguajes Formales.

Objetivos

- Emplear los REGEX para los sufijos y prefijos, en la disminución de las palabras de un texto.
- Obtener un resultado claro de cuantas palabras se repetían al momento de aplicar el algoritmo de Porter

Resumen

La frecuencia de una palabra en un texto puede ser útil para muchas tareas, por ejemplo, para clasificación de textos o clustering, recuperación de información, generación de resúmenes, etc.

He ahí donde radica la importancia del algoritmo de Porter, pues este algoritmo asegura que la forma de las palabras no penalice la frecuencia de éstas. Es decir, una palabra puede estar conjugada en cualquier género, número, persona, etc… y solo se considerará (en muchas ocasiones) como un solo término.

El presente trabajo consiste en una adaptación del algoritmo de Porter para procesar documentos en español.

Introducción

En un documento una palabra dada puede ser encontrada de muchas formas diferentes. Por ejemplo, las palabras, "computadora", "computador", "computacional", "computando", "Computadoras", "Computadores", junto con algunas otras palabras, tienen diferente forma, pero todas hacen referencia cercana a un mismo concepto. Si algún usuario requiriera información sobre Computadoras podría perderse de encontrar documentos relevantes, sólo porque estas palabras están escritas de otra forma y tienen poca frecuencia en el documento, por ende, no se tomaron en cuenta como palabras índices. Esto es claramente indeseable.

Una manera de solucionar este problema es introducir algoritmos de stemming, los cuales eliminan las terminaciones de las palabras, reduciéndolas a un término común o raíz (stem). En el ejemplo presentado, la raíz sería "Comput". Para un documento dado, esto podría unir varias palabras con la misma raíz y, de esta manera, elevar su frecuencia en el texto, haciéndolas candidatas a término común.

Numerosos algoritmos de stemming se vienen desarrollando desde hace años. Tres de los más conocidos son los construidos por Lovins en 1968, Porter en 1980 y Paice en 1990. Todos estos algoritmos, van eliminando consecutivamente los finales de las palabras, para arribar a su raíz.

Algoritmo de Porter

El algoritmo de Porter nos permite realizar stemming, esto es remover los sufijos comunes morfológicos e inflexionales de palabras literalmente diferentes, pero con un stem común, que pueden ser consideradas como un sólo término. Este algoritmo requiere de un conjunto de pasos para llegar al stem.

Porter publicó en 1980 un algoritmo para el método de Stemming que fue tomado como base por muchos investigadores. El algoritmo lee un archivo, toma una serie de caracteres y de esa serie, una palabra; luego la valida verificando que todos los caracteres involucrados sean letras, de ser así, aplica Stemming sobre ella.

La aplicación de Stemmer consiste en hacer pasar esta palabra a través de varios conjuntos de reglas, cada conjunto de reglas está formada por n reglas y cada regla por:

1. Un identificador de regla
2. El sufijo a identificar
3. El texto por el cual debe ser reemplazado al encontrar el sufijo
4. El tamaño del sufijo
5. El tamaño del texto de reemplazo
6. El tamaño mínimo que debe tener la raíz resultante luego de aplicar la regla (esto es a los efectos de no procesar palabras demasiado pequeñas)
7. Una función de validación (una función que verifica si se debe aplicar la regla una vez encontrado el sufijo)

Considere, a modo de ejemplo, la siguiente regla perteneciente al algoritmo original de Porter:

106, "ed", LAMBDA, 1, -1, -1, ContainsVowel

Analizándola elemento a elemento se tiene que:

1. 106 es el identificador de la regla.
2. "ed" es el sufijo que debe localizar al final de la palabra.
3. "LAMBDA" es el texto por el cual se debe reemplazar el sufijo una vez encontrado (en este caso LAMBDA es una constante definida como cadena vacía).
4. "1" es el tamaño del sufijo.
5. "-1" es el tamaño del texto que se debe reemplazar por el sufijo, como se trata, en este caso, de una cadena vacía el tamaño sería 0 que restándole 1 sería –1.
6. El siguiente "–1" es el tamaño mínimo que debe tener la raíz una vez que le quitemos "ed" (-1significa que lo quite cuando al menos queden tres caracteres).
7. "ContainsVowels" es la función de validación. En el caso particular de ContainsVowels, verifica que la palabra sin "ed" contenga vocales.

Por lo cual esta regla se aplicaría a las palabras lowed quedando low, shared quedando shar, pero no se aplicaría a la palabra shed, ya que si le sacáramos “ed” no quedarían vocales y además, la raíz quedaría con 2 letras y se pide que tenga al menos 3.

Marco histórico del Algoritmo

El documento original del algoritmo de stemming se escribió en 1979 en el Laboratorio de Computación, Cambridge (Inglaterra), como parte de un proyecto de IR más grande, y apareció como Capítulo 6 del informe final del proyecto:

C.J. van Rijsbergen, S.E. Robertson y M.F. Porter, 1980. *Nuevos modelos en recuperación de información probabilística*. Londres: Biblioteca Británica. (Informe de Investigación y Desarrollo de la Biblioteca Británica, n. 5587).

Con el estímulo de van Rijsbergen, también se publicó en

M.F. Porter, 1980, *Un algoritmo para la eliminación de sufijos, Programa*, 14 (3) pp 130-137.

Y desde entonces ha sido reimpreso en.

Karen Sparck Jones y Peter Willet, 1997, *Readings in Information Retrieval*, San Francisco: Morgan Kaufmann, ISBN 1-55860-454-4.

El stemmer original fue escrito en BCPL, un lenguaje que alguna vez fue popular, pero que ahora está extinguido. Durante los primeros años posteriores a 1980, se distribuyó en su forma BCPL, a través de una cinta de papel perforada. Pronto empezaron a aparecer versiones en otros idiomas, y para 1999 se estaba utilizando, citando y adaptando ampliamente. Desafortunadamente, hubo numerosas variaciones en la funcionalidad entre estas versiones, y esta página web se configuró principalmente para "poner las cosas en orden" y establecer una versión definitiva para la distribución.

Aplicaciones del Algoritmo de Porter

- Clasificación de textos
- Recuperación de información
- Generación de resúmenes

Minería de datos

La minería de datos es el proceso de detectar la información procesable de los conjuntos grandes de datos. Utiliza el análisis matemático para deducir los patrones y tendencias que existen en los datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque hay demasiado datos.

Estos patrones y tendencias se pueden recopilar y definir como un *modelo de minería de datos*. Los modelos de minería de datos se pueden aplicar en escenarios como los siguientes:

- **Previsión**: Calcular las ventas y predicción de las cargas del servidor o el tiempo de inactividad del servidor
- **Riesgo y probabilidad**: Elegir a los mejores clientes para la distribución de correo directo, determinación del punto de equilibrio probable para escenarios de riesgo, y asignar probabilidades a diagnósticos y otros resultados
- **Recomendaciones**: Determinar qué productos se pueden vender juntos y generación de recomendaciones
- **Buscar secuencias**: Análisis de cliente en un carro de la compra, predicción de posibles eventos probables
- **Agrupación**: Distribución de clientes o eventos de elementos relacionados, analizar y predecir afinidades en

La tarea de minería de datos real es el análisis automático o semiautomático de grandes cantidades de datos para extraer patrones interesantes hasta ahora desconocidos, como los grupos de registros de datos (análisis clúster), registros poco usuales (la detección de anomalías) y dependencias (minería por reglas de asociación). Esto generalmente implica el uso de técnicas de bases de datos como los índices espaciales. Estos patrones pueden entonces ser vistos como una especie de resumen de los datos de entrada, y pueden ser utilizados en el análisis adicional o, por ejemplo, en el aprendizaje automático y análisis predictivo. Por ejemplo, el paso de minería de datos podría identificar varios grupos en los datos, que luego pueden ser utilizados para obtener resultados más precisos de predicción por un sistema de soporte de decisiones.

Minería Web

El Web mining o minería web es una metodología de recuperación de la información que usa herramientas de la minería de datos para extraer información tanto del contenido de las páginas, de su estructura de relaciones (enlaces) y del registro de navegación de los usuarios.

es la aplicación de técnicas de minería de datos para descubrir los patrones de la Web. De acuerdo a los objetivos de análisis, la minería web se puede dividir en tres tipos diferentes, que son la minería de uso de la Web, minería del contenido de la Web y minería de la estructura de la Web.

En este sentido podemos definir la minería Web en tres variantes:

1. Minería del contenido de la Web, o *Web Content Mining*;
2. Minería de la estructura de la Web, o *Web Structure Mining*;
3. Minería de los registro de navegación en la Web, o *Web Usage Mining*.

- Minería del uso de la Web:

La minería del uso de la Web es un proceso de extracción de información útil a partir de los registros del servidor, es decir, del historial de los usuarios.

La minería del uso de la Web es el proceso de descubrir lo que los usuarios buscan en Internet. Algunos usuarios pueden estar mirando sólo los datos textuales, mientras que otros pueden estar interesados en los datos multimedia.

- Minería del contenido de la Web:

Minería del contenido de la Web es el proceso de descubrir información útil de texto, imagen, audio o datos de vídeo en la web.

La minería de contenido web a veces se llama la minería de textos web, porque el contenido del texto es la zona más ampliamente investigado. Las tecnologías que se utilizan normalmente en la minería de contenido web son PLN (procesamiento de lenguaje natural) y RI (recuperación de información).

- Minería de la estructura de la Web:

Minería de la estructura de la Web es el proceso de utilización de la teoría de grafos para analizar el nodo y la estructura de conexión de un sitio web.

Según el tipo de web de los datos estructurales, estructura de minería de la Web se pueden dividir en dos tipos:

- El primer tipo es la extracción de patrones a partir de hipervínculos de la web. Un hipervínculo es un componente estructural que conecta a la página web en una ubicación diferente.
- El otro tipo es la minería de la estructura del documento. Se está utilizando la estructura de árbol para analizar y describir el HTML (Hyper Text Markup Language) o XML (eXtensible Markup Language) tags dentro de la página web.

Minería de Texto

La minería de textos busca extraer información útil e importante de formatos de documentos heterogéneos, tales como páginas web, correos electrónicos, medios sociales, artículos de revistas, etc. Esto se hace mediante la identificación de patrones dentro de los textos, tales como tendencias en el uso de palabras, estructura sintáctica, etc.

La minería de textos tiene muchas aplicaciones. Por ejemplo, la minería de textos puede ayudar a encontrar tecnologías nuevas e innovadoras dentro de ciertos dominios. Es un método muy eficiente para generar nueva información y conocimiento. Esta práctica permite a las empresas reducir el tiempo dedicado a la lectura de textos extensos y extractos literarios. Esto significa que los recursos clave se pueden encontrar con mayor rapidez y eficacia. También permite a los usuarios obtener nueva información que de otro modo sería difícil de encontrar.

La tecnología de la minería de textos es ampliamente aplicada en la actualidad por una extensa variedad de usuarios, desde organizaciones gubernamentales, instituciones de investigación y empresas para sus necesidades diarias. Estos son algunos ejemplos de uso en diferentes campos:

- ***Investigación*:** por ejemplo, el descubrimiento de conocimientos, la atención médica y sanitaria; en el pasado, a un investigador humano le lleva mucho tiempo analizar y obtener información relevante. En algunos casos, esta información ni siquiera era accesible. La minería de textos permite a los investigadores encontrar más información y de forma más rápida y eficiente.

- ***Negocios*:** por ejemplo, las grandes empresas utilizan la minería de textos para ayudar en la toma de decisiones y responder rápidamente a las consultas de los clientes en procesos tales como la gestión de riesgos o el filtrado de currículos

- ***Seguridad***: En anti-terrorismo, el análisis de los blogs y otras fuentes de texto en línea se utiliza para prevenir delitos en Internet y luchar contra el fraude.

Diariamente, la minería de texto es usada por los sitios web de correo electrónico para crear métodos de filtrado más confiables y efectivos, para el filtrado de spam, análisis de datos de medios sociales, etc. También para identificar las relaciones entre los usuarios y ciertos productos o para determinar las opiniones de los usuarios sobre temas particulares.

La extracción de textos puede dividirse en cinco pasos:

1. ***Recolección***: Recopilación de datos de diferentes recursos, tales como sitio web, correos electrónicos, comentarios de clientes, archivo de documentos. Dependiendo de la aplicación, este proceso puede ser completamente automatizado o guiado por una persona encargada de realizar este proceso.

2. ***Preprocesamiento****:* La identificación del contenido y la extracción de características representativas

3. ***Limpieza de textos***: eliminación de cualquier información innecesaria o no deseada, como los anuncios de las páginas.

4. ***Tokenización*****:** un ordenador sólo "ve" una cadena de caracteres, sin poder identificar, por ejemplo, párrafos, frases o palabras. La *Tokenización* divide el texto en entidades significativas (palabras, oraciones, etc.) dados los espacios en blanco presentes y las puntuaciones.

5. ***Extracción de característica:*** es el proceso de caracterización.

Autómatas (Avance 1)

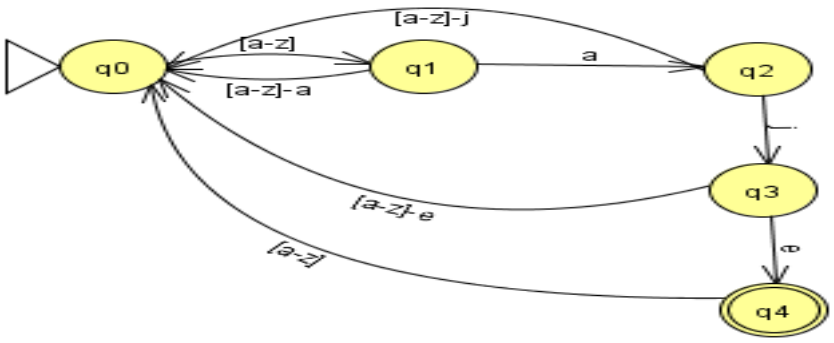
Un autómata es un modelo matemático para una máquina de estado finito. Una máquina de estado finito es una máquina que, dada una entrada de símbolos, "salta" a través de una serie de estados de acuerdo a una función de transición (que puede ser expresada como una tabla).

La entrada es leída símbolo por símbolo, hasta que es "consumida" completamente (piense en ésta como una cinta con una palabra escrita en ella, que es leída por una cabeza lectora del autómata; la cabeza se mueve a lo largo de la cinta, leyendo un símbolo a la vez) una vez la entrada se ha agotado, el autómata se detiene.

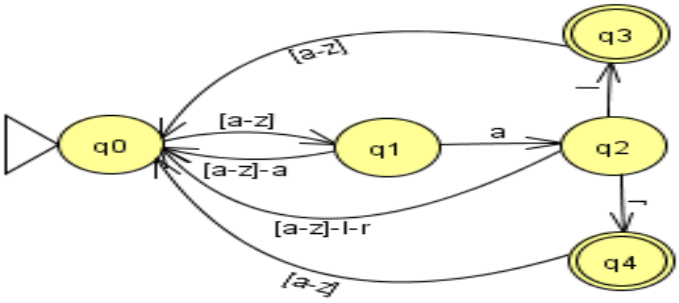
Dependiendo del estado en el que el autómata finaliza se dice que este ha aceptado o rechazado la entrada. Si éste termina en el estado "acepta", el autómata acepta la palabra. Si lo hace en el estado "rechaza", el autómata rechazó la palabra, el conjunto de todas las palabras aceptadas por el autómata constituyen el lenguaje aceptado por el mismo.

Para este trabajo se crearon distintos autómatas específicos para cada tipo de sufijos y prefijos. A continuación, algunos ejemplos de estos:

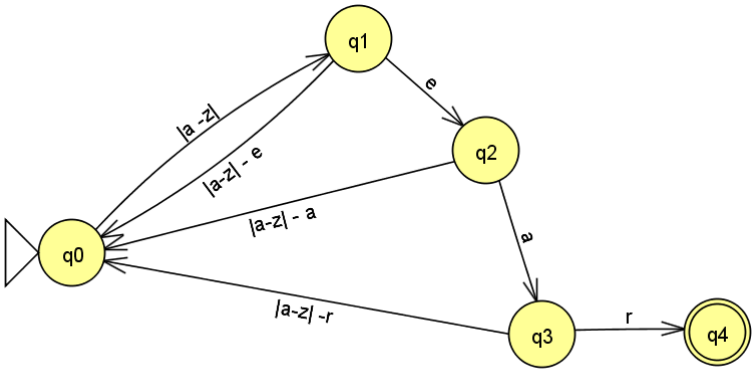
<u>Autómata para SUFIJOS de SUSTANTIVOS “-aje”:</u>



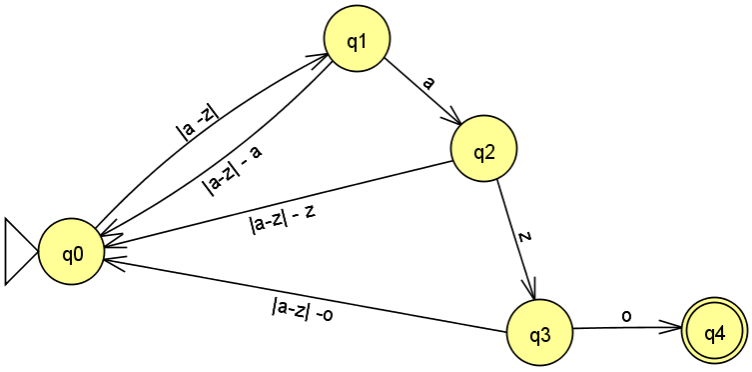
Autómata para SUFIJOS de ADJETIVOS “-al, -ar”:



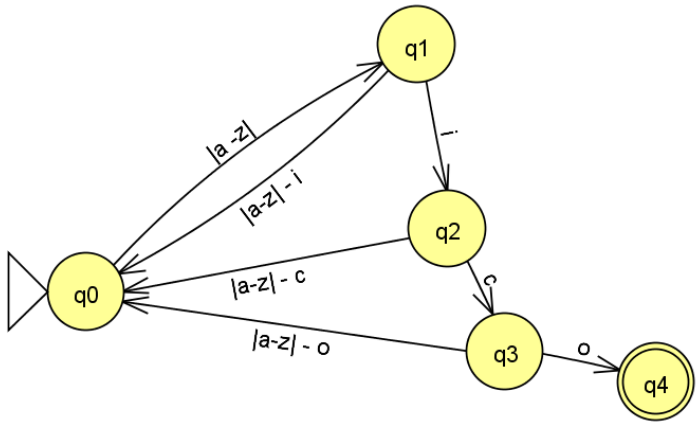
Autómata para SUFIJOS de VERBOS “-ear”:



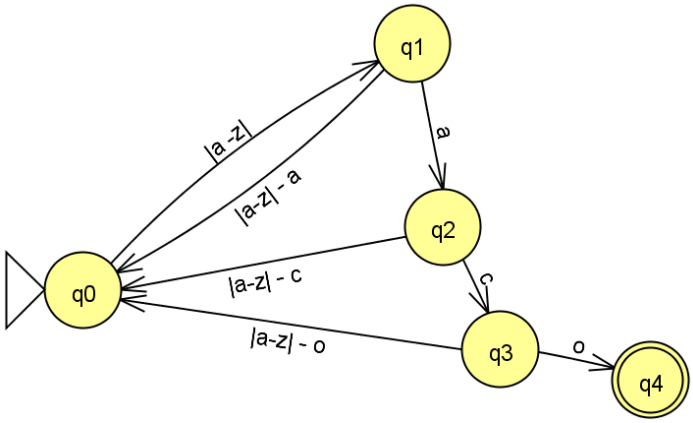
Autómata para SUFIJOS AUMENTATIVOS “-azo”:



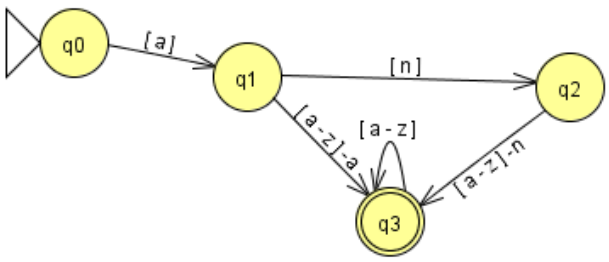
Autómata para SUFIJOS DIMINUTIVOS “-ico”:



Autómata para SUFIJOS DESPECTIVOS “-aco”:



Autómata para PREFIJOS:



Expresiones Regulares (Avance 2)

Una expresión regular, también conocida como REGEX, es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

En informática, las expresiones regulares proporcionan una manera muy flexible de buscar o reconocer cadenas de texto.

Su utilidad más obvia es la de describir un conjunto de cadenas para una determinada función, resultando de utilidad en editores de texto y otras aplicaciones informáticas para buscar y manipular textos.

Numerosos editores de texto y otras herramientas utilizan expresiones regulares para buscar y reemplazar patrones en un texto.

Los siguientes métodos pueden implementarse con expresiones regulares:

- Coincidencia de patrones
- Globbing
- Truncation
- Stemming

Los patrones regex están formados por elementos particulares, letras, símbolos, números o combinaciones de estos, que indican las condiciones que se deben cumplir en la cadena para considerarla una coincidencia.

Algunos de estos elementos son:

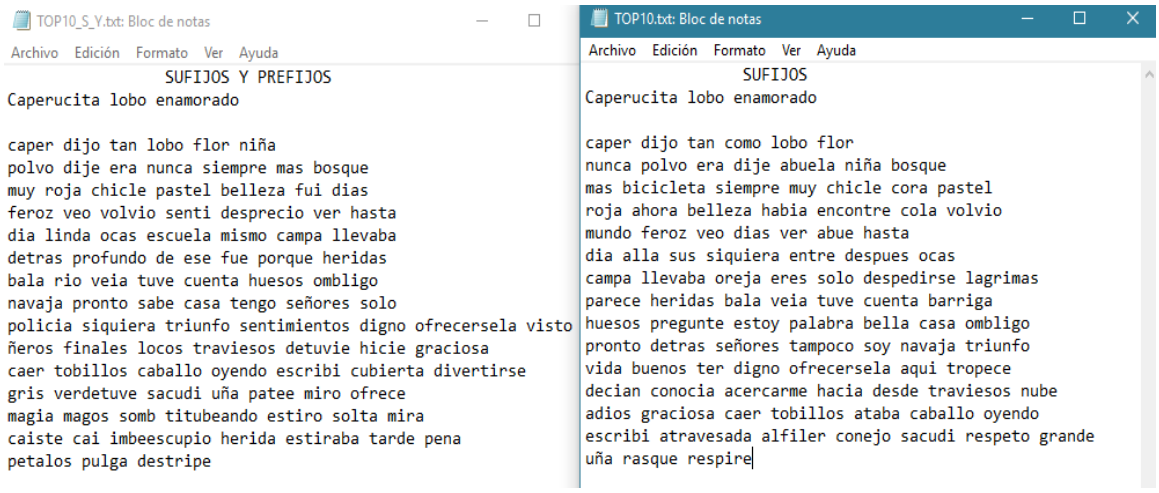
- Los corchetes **[ ]**
- Los paréntesis **( )**
- La barra invertida \
- El pipe **|**
- Y algunos símbolos ***** y **?**

Ejemplos de una sentencia en Java

```
Pattern P_pronombre7 =Pattern.compile(PRONOMBRE7);
COINCIDE = P_pronombre7.matcher(LineaTexto);
LineaTexto = COINCIDE.replaceAll("");
```

Comparación de métodos

A continuación, se presenta una comparación entre el método de los sufijos y prefijos contra el de los sufijos:



Como se ve en el texto, al aplicar cada uno de estos métodos cada palabra del texto que contenga un sufijo o un prefijo queda en su estado raíz, es decir, sin sufijos o prefijos.

Ahora, ¿Cuál de estos métodos es más eficiente?

Si hablamos en cuanto a dejar una palabra en una raíz, se puede decir que es mas eficiente el primer método (Sufijos y Prefijos), ya que, al dejar las palabras en un estado mas generalizado, puede abarcar mas palabras que contengan dicha raíz, pero en cuanto a el análisis semántico del texto resumen, el segundo método seria el mas indicado, ya que al no generalizar tanto la raíz deja estas a un alcance mayor al entendimiento del usuario.

Conclusiones

El algoritmo de Porter es muy eficiente a la hora de realizar resúmenes, cuentas de coincidencias y distintas otras tareas, pero su principal defecto es que fue originalmente pensado y desarrollado para la lengua inglesa. A pesar de que ya existe su debida versión para el español, se siguen presentando problemas para este a la hora de eliminar algunos verbos irregulares o artículos específicos.

Al momento de implementar este algoritmo se dio a notar la falta de un procesador de lenguaje natural, ya que este podría interpretar la gran cantidad de palabras en su forma raíz para luego entregar un texto que le permita al usuario comprender de manera más fácil las ideas del texto trabajado.