

Variantes de TextRank para la Generación de Resúmenes Automáticos

Federico Barrios, Federico López, Luis Argerich, Rosa Wachenchauzer

Facultad de Ingeniería, Universidad de Buenos Aires,
Av. Paseo Colón 850, Ciudad Autónoma de Buenos Aires, Argentina
`{fbarrios,fjlopez}@fi.uba.ar`
<http://ingenieria.uba.ar/>

Resumen Este reporte describe el desarrollo y prueba de nuevas variantes del algoritmo TextRank para la generación de resúmenes automáticos. Se describen las generalidades del algoritmo TextRank en su versión original y las diferentes variantes del algoritmo que fueron probadas. Algunas de estas variantes logran una mejora en la performance del algoritmo de acuerdo a las métricas standard para la evaluación de resúmenes automáticos.

Keywords: TextRank, resumen, automático

1. Introducción

El proceso de generación de resúmenes automáticos consiste en extraer de un documento las oraciones mas representativas del mismo. Con diferentes niveles de compresión puede obtenerse una representación resumida del documento de una longitud arbitraria. El algoritmo TextRank es uno de los más utilizados para esa tarea y se basa en la generación de puntajes para cada oración del texto basado en el resultado de aplicar el conocido algoritmo PageRank al grafo generado por las diferentes oraciones del texto. Se pueden lograr variantes del algoritmo utilizando diferentes propuestas para el armado del grafo. En este reporte se describen cuáles fueron las variantes probadas y sus resultados.

2. Trabajo previo

Se ha visto un gran avance en el campo de la generación automática de resúmenes desde finales de 1960 hasta la actualidad [1]. Los métodos tradicionales tienen en cuenta la frecuencia de palabras o frases introductorias para identificar las oraciones más sobresalientes del texto. Asimismo, se han desarrollado modelos estadísticos basados en corpus de entrenamiento para combinar diferentes heurísticas: palabras clave, posición y longitud de las oraciones, frecuencia de palabras y contenido de los títulos [2]. Otros enfoques se basan en la representación del texto en forma de grafo: las oraciones importantes y los conceptos son las entidades altamente conectadas y, por esto, forman parte del

resumen [3]. De igual modo, se ha propuesto analizar la estructura discursiva y extraer las relaciones retóricas entre las diferentes unidades textuales, y así separar las principales de las secundarias para descubrir aquéllas que juegan un papel preponderante [4].

En la línea de representación del texto como un grafo conectado se usan técnicas de Recuperación de Información para identificar oraciones similares y determinar las más importantes, que formarán al resumen final [5]. El enfoque propuesto, tanto por Mihalcea y Tarau como por Erkan y Radev [6], consiste en utilizar el prestigio de las unidades léxicas (oraciones o palabras) dentro del grafo. Dicha técnica es la que adopta el algoritmo de TextRank.

3. TextRank

3.1. Descripción

TextRank es un algoritmo no supervisado basado en grafos para realizar resúmenes automáticos extractivos u obtener palabras claves de un texto. Fue presentado en 2004 por Rada Mihalcea y Paul Tarau en la publicación “TextRank: Bringing Order into Texts” [7].

El algoritmo aplica una variación de PageRank [8] sobre un grafo especialmente diseñado para la tarea. De esta manera permite explotar la estructura del texto, identificando los conceptos principales, sin necesidad de datos previos de entrenamiento. Debido a que se basa en PageRank, se sirve de la noción del “prestigio” o “recomendación” entre los elementos del grafo, lo que lo hace aplicable a cualquier texto incluso sin importar el idioma.

El algoritmo modela el texto en base a un grafo, y luego busca crear relaciones significativas (aristas) entre las entidades léxicas (vértices). Dependiendo de la aplicación que se desee dar al algoritmo, las entidades pueden ser palabras, frases, oraciones, párrafos, entre otros. De manera similar, también debe definirse el tipo de relación que se usa para unir los vértices: semántica, contextual, de superposición, y demás.

Los pasos principales que se llevan a cabo son los siguientes:

1. Identificar las unidades del texto (palabras u oraciones) y agregarlas al grafo como vértices.
2. Identificar relaciones que conectan a estas unidades, y agregarlas al grafo como aristas entre los vértices. Las aristas pueden ser dirigidas o no, y ponderadas o no.
3. Aplicar PageRank para asignarle un puntaje a cada vértice.
4. Ordenar los vértices de acuerdo al puntaje y utilizarlo para armar el resumen de acuerdo a algún criterio.

3.2. Generación de resúmenes automáticos

El problema de la extracción de oraciones apunta a identificar las secuencias más representativas del texto. Para este caso, las unidades tomadas para aplicar el algoritmo serán oraciones completas [9].

Inicialmente, se construye un grafo en base al texto. Cada oración se considera como un vértice, y para asignarles peso a las aristas se debe definir una función de similitud entre dos oraciones dadas. Esta función será la que dicte cuánto una oración “recomienda” a otra, dependiendo de la similitud entre sus contenidos por abordar los mismos conceptos.

La función utilizada se define formalmente de la siguiente manera:

Definición 1 Sean S_i, S_j dos oraciones representadas por un conjunto de n palabras que en S_i aparecen como $S_i = w_1^i, w_2^i, \dots, w_n^i$. La función de similitud para S_i, S_j se define como:

$$Similitud(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (1)$$

El resultado de este proceso es el texto representado como un grafo ponderado y altamente denso. En base a esto se aplica PageRank para calcular la relevancia de cada vértice.

Luego de aplicar el algoritmo de priorización se seleccionan las oraciones con mayor puntaje para incluirlas en el resumen. Estas, finalmente, se presentan de acuerdo al orden de aparición en el texto original.

4. Experimentos

4.1. Evaluación

Se decidió usar la base de datos de la tarea de generación de resúmenes automáticos de la conferencia DUC (Document Understanding Conference) del año 2002, al igual que en la publicación original del algoritmo TextRank. El corpus cuenta de 567 documentos que fueron resumidos a cerca del 20 %.

Para llevar a cabo la evaluación se usó la versión 1.5.5 del paquete de métricas ROUGE [10]. Se utilizó la configuración usada en DUC, calculando sólo las mediciones ROUGE-1, ROUGE-2 y ROUGE-SU4 en un intervalo de confianza del 95 % y aplicando un método de *stemming*. Se obtuvo un único puntaje promediando los tres valores.

Para verificar el funcionamiento de todo el conjunto se implementó el método de referencia (*baseline*) usado en la publicación, que construye el resumen extrayendo las primeras oraciones de cada artículo. Se comprobó que los resultados fueron similares: la versión original de TextRank mejora al *baseline* por cerca de un 2,3 %.

4.2. Propuestas

En la siguiente sección se describen las variantes que se probaron sobre el algoritmo original de TextRank. Principalmente estas modificaciones apuntan a cambiar la forma en la cual se calculan las distancias entre las oraciones de los textos. Diferentes métricas arrojaron diferentes resultados y algunas de éstas variantes lograron una mejora significativa sobre el algoritmo original.

Subcadena común Dadas dos frases, el problema de la subcadena común más larga consiste en identificar la secuencia de caracteres de mayor extensión presente en ambas. Por ejemplo, entre “la cocina verde” y “una cocina vale más si es verde”, la secuencia más larga es “a cocina v”.

La propuesta consiste en modificar la función de distancia de la implementación original y reemplazarla por el largo de la subcadena común más larga. En el ejemplo anterior, la similitud sería de 10, dado que es el largo de la secuencia “a cocina v”.

Cabe destacar el parecido de esta métrica con los métodos de evaluación ROUGE.

Similitud coseno La similitud coseno es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Para poder hacer uso de esta “distancia” se utiliza el modelo del espacio vectorial, y así modelar a los documentos como vectores.

Este modelo algebraico es utilizado para representar documentos en lenguaje natural de una manera formal mediante el uso de vectores en un espacio lineal multidimensional. La teoría básica es que el parecido de un documento frente a otro puede calcularse usando la similitud coseno. Así un valor de coseno de cero significa que los documentos son ortogonales el uno al otro, y eso significa que no hay similitud.

La propuesta se basa en aplicar este modelo para tratar a cada oración del texto como un vector n-dimensional (con n la cantidad de palabras distintas presentes en el documento), y luego compararlas utilizando la similitud coseno. Las entradas de cada vector estarán compuestas por el resultado de aplicar la función “frecuencia de término – frecuencia inversa de documento” (TF-IDF de sus siglas en inglés) a cada palabra de la oración representada.

Tomando como ejemplo el documento “Esta bien. Todo bien.”, las oraciones quedarían modeladas como vectores de la siguiente forma:

$$\begin{aligned} v1 &= [\text{TFIDF}(\text{"Esta"}) , \text{TFIDF}(\text{"bien"}) , 0] \\ v2 &= [0 , \text{TFIDF}(\text{"bien"}) , \text{TFIDF}(\text{"Todo"})] \end{aligned}$$

Dado que la imagen de la función TFIDF está contenida en el intervalo [0,1], todos los vectores quedan conformados por entradas no negativas, haciendo que ninguna similitud sea menor a cero.

BM25 Okapi BM25 es una función de ranking utilizada para la asignación de relevancia a los documentos en un buscador. Está basada en los modelos probabilísticos de Recuperación de información, y actualmente representa el estado del arte en algoritmos de recuperación de documentos basados en frecuencia de término – frecuencia inversa de documento.

Definición 2 Dadas dos oraciones R , S , BM25 se define como:

$$BM25(R, S) = \sum_{i=1}^n IDF(s_i) \cdot \frac{f(s_i, R) \cdot (k_1 + 1)}{f(s_i, R) + k_1 \cdot (1 - b + b \cdot \frac{|R|}{avgDL})} \quad (2)$$

donde k y b son parámetros que valen $k = 1,2$ y $b = 0,75$, y $avgDL$ es el largo promedio de las oraciones en el texto.

Esta función contempla la frecuencia inversa del documento de manera tal que si una palabra aparece en más de la mitad de los textos, el valor para ese término se vuelve negativo. Dado que este comportamiento no es deseable para esta aplicación, se utiliza la siguiente fórmula:

$$IDF(q_i) = \begin{cases} \log(N - n(q_i) + 0,5) - \log(n(q_i) + 0,5) & \text{si } n(q_i) > N/2 \\ \varepsilon \cdot avgIDF & \text{si } n(q_i) \leq N/2 \end{cases} \quad (3)$$

donde ε ronda entre 0,5 y 0,30 y $avgIDF$ es el idf promedio para todos los términos. También se ensayaron alternativas como igualar a cero el resultado en caso de ser negativo, o reemplazar la función inversa por una similar, cuyo resultado es no negativo, o estrictamente positivo.

Existe una variante a BM25 llamada BM25+. Esta repara deficiencias relacionadas a la frecuencia de término y a la penalización a documentos largos frente a documentos cortos irrelevantes.

La propuesta consistió en aplicar BM25 y BM25+ entre las oraciones de un texto, a fin de ordenarlas de acuerdo a su relevancia.

4.3. Resultados

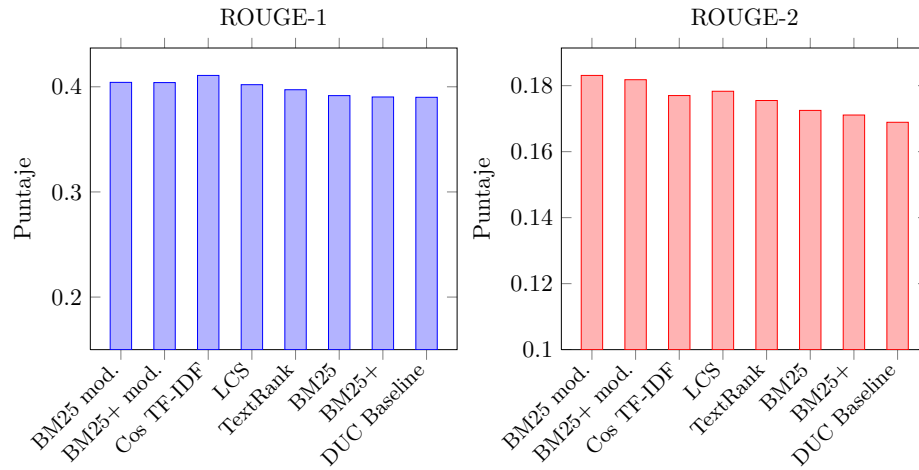
Los mejores resultados se obtuvieron usando BM25 y BM25+. El incremento más alto se logró al reemplazar los valores negativos por la constante $\varepsilon = 0.25$, dando una mejora total de 2,92 % para BM25 y 2,60 % para BM25+. Inicialmente se obtuvieron valores cercanos al 1,30 %, pero dado que esta función contempla el uso de *stopwords* se procedió a eliminar la etapa de filtrado del preprocesamiento del texto, alcanzando así resultados sobresalientes. En el Cuadro 1 se detallan las alternativas ensayadas.

Los tiempos de ejecución también se superaron. Se pudo procesar los 567 documentos de la base de datos de DUC2002 utilizando 84 % del tiempo requerido por la versión original.

El resultado de la métrica de similitud coseno también fue satisfactoria, presentando una mejora de 2,54 % por sobre el método original. A su vez, los métodos de secuencias de máxima longitud también mostraron una mejora considerable: cerca del 1,40 % por sobre TextRank. En la Figura 1 se comparan los distintas técnicas.

Cuadro 1. Resultados de las distintas propuestas

Método	ROUGE-1	ROUGE-2	ROUGE-SU4	Mejora
BM25 (Neg a epsilon)	0.4042	0.1831	0.2018	2,92 %
BM25+ (Neg a epsilon)	0.404	0.1818	0.2008	2,60 %
Cosine TF-IDF	0.4108	0.177	0.1984	2,54 %
BM25+ (IDF = $\log(N/NI)$)	0.4022	0.1805	0.1997	2,05 %
BM25 (IDF = $\log(N/NI)$)	0.4012	0.1808	0.1998	1,97 %
Longest Common Substring	0.402	0.1783	0.1971	1,40 %
BM25+ (Neg a cero)	0.3992	0.1803	0.1976	1,36 %
BM25 (Neg a cero)	0.3991	0.1778	0.1966	0,89 %
TextRank	0.3983	0.1762	0.1948	—
BM25	0.3916	0.1725	0.1906	-1,57 %
BM25+	0.3903	0.1711	0.1894	-2,07 %
DUC Baseline	0.39	0.1689	0.186	-2,84 %

**Figura 1.** Comparación de métricas.

5. Conclusiones

En este trabajo se analizaron variantes al algoritmo de TextRank. A partir del mismo se propusieron e implementaron optimizaciones cuyos resultados fueron significativos: se obtuvo una mejoría del 2,92 % por sobre el método original. Este número es notable si se tiene en cuenta que TextRank por sí solo performa 2,84 % por sobre el estándar de comparación (baseline).

Las evaluaciones fueron hechas con las mismas técnicas y conjuntos de datos utilizados en competencias internacionales, validando así las mediciones.

Queda para próximos trabajos explorar alternativas para la extracción de palabras claves, y el ensayo de distintos métodos algebraicos, haciendo uso del modelo del espacio vectorial.

Referencias

1. Sabino, Miranda Jiménez: *Modelos para la generación automática de resúmenes abstractivos basado en grafos conceptuales*. Tesis de Doctorado, Instituto politécnico nacional, August 2013.
2. Lin, Chin Yew y E. H. Hovy: *Identifying Topics by Position*. En *Proceedings of 5th Conference on Applied Natural Language Processing*, Washington D.C., March 1997.
3. Barzilay, Regina y Kathleen McKeown: *Sentence Fusion for Multidocument News Summarization*. Computational Linguistics, 31(3):297–328, 2005. <http://dblp.uni-trier.de/db/journals/coling/coling31.html#BarzilayM05>.
4. Marcu, Daniel: *The Theory and Practice of Discourse Parsing and Summarization*. Computational Linguistics, 28(1):81–83, 2000. <http://dblp.uni-trier.de/db/journals/coling/coling28.html#Hahn02>.
5. Salton, G., A. Singhal, M. Mitra y C. Buckley: *Automatic Text Structuring and Summarization*. Information Processing and Management, 33(2):193 – 207, 1997.
6. Erkan, Günes y Dragomir R. Radev: *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization*. J. Artif. Intell. Res. (JAIR), 22:457–479, 2004. <http://dblp.uni-trier.de/db/journals/jair/jair22.html#ErkanR04>.
7. Mihalcea, Rada y Paul Tarau: *TextRank: Bringing Order into Texts*. En Lin, Dekang y Dekai Wu (editores): *Proceedings of EMNLP 2004*, páginas 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.
8. Page, L., S. Brin, R. Motwani y T. Winograd: *The PageRank citation ranking: Bringing order to the Web*. En *Proceedings of the 7th International World Wide Web Conference*, páginas 161–172, Brisbane, Australia, 1998. citeseer.nj.nec.com/page98pagerank.html.
9. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: *Introduction to Information Retrieval*. Cambridge University Press, 2008.
10. Lin, Chin Yew: *ROUGE: a Package for Automatic Evaluation of Summaries*. En *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, 2004.