Variantes de TextRank para la Generación de Resúmenes Automáticos

Resumen Este artículo 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 que fueron propuestas y probadas. Algunas de estas logran una mejora en la performance del algoritmo de acuerdo a las métricas estándar para la evaluación de resúmenes automáticos.

Keywords: TextRank, resumen, automático

1. Introducción

El proceso de generación extractiva de resúmenes automáticos consiste en extraer de un documento sus oraciones más representativas. Con diferentes niveles de compresión puede obtenerse una representación resumida del documento de una longitud arbitraria. El algoritmo TextRank se basa en aplicar el conocido algoritmo PageRank al grafo generado por las diferentes oraciones del texto. En este trabajo se describen variantes al algoritmo probadas, y sus resultados.

La primera sección de este artículo describe el trabajo previo existente en el área y el algoritmo TextRank en general. Luego se detallan las distintas variantes propuestas al algoritmo original y los resultados obtenidos con las mismas. Finalmente se describen las métricas utilizadas para la evaluación de los cambios propuestos y los conjuntos de datos utilizados.

2. Trabajo previo

Se ha visto interés en el campo de la generación automática de resúmenes desde finales de la década del 1950 [1]. Los métodos tradicionales tienen en cuenta la frecuencia de palabras o frases introductorias para identificar las oraciones más sobresalientes del texto. Otros enfoques más modernos 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 [2].

Con esta idea se usan métodos del campo de Recuperación de Información para identificar oraciones similares y determinar las más importantes, que formarán al resumen final [3]. El enfoque propuesto, tanto por Mihalcea y Tarau en TextRank [4] como por Erkan y Radev en LexRank [5], consiste en utilizar el prestigio de las unidades léxicas (oraciones o palabras) dentro del grafo.

3. TextRank

TextRank es un algoritmo no supervisado para generar resúmenes automáticos u obtener palabras claves de un texto. Fue presentado en 2004 por Rada Mihalcea y Paul Tarau [4].

El algoritmo modela el texto en base a un grafo, intentando crear relaciones significativas (aristas) entre las oraciones del texto (vértices). A este grafo se le aplica una variación de PageRank [6], sirviéndose de la noción de "recomendación" entre las unidades léxicas. De esta manera explota la estructura del texto y determina la importancia de sus oraciones sin necesidad de datos previos de entrenamiento. Este método, en consecuencia, es aplicable a cualquier texto, incluso sin importar el idioma.

El peso de cada arista queda determinado por una función de semejanza entre los vértices que conecta. 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 por el algoritmo 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 $S_i = w_1^i, w_2^i, ..., w_n^i$. 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. Luego de aplicar el algoritmo de PageRank 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. Variantes propuestas

Las variantes que se proponen al algoritmo original de TextRank consisten en cambiar la forma en la que se calcula la similitud entre las oraciones de los textos. En esta sección se describen aquellas con las que se obtuvieron mejores resultados.

4.1. Subcadena común

El problema de la subcadena común más larga consiste en identificar la secuencia de caracteres de mayor extensión presente en dos oraciones. 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 reemplazar la función de similitud por el largo de la subcadena común más larga. En el ejemplo anterior, esta longitud sería de 10, dado que es el largo de la secuencia "a cocina v".

4.2. Similitud coseno

La similitud coseno es una medida del parecido entre dos vectores en un espacio que posee un producto interior, y resulta de evaluar el valor del coseno del ángulo comprendido entre ellos. Para poder hacer uso de esta "distancia" se utiliza el modelo vectorial, es decir, se modelan a los documentos como vectores.

Este modelo algebraico es utilizado para representar documentos en lenguaje natural de una manera formal, y se basa en favorecer la dirección a la cuál apuntan los documentos independientemente de su longitud. Consecuentemente, textos que hablan de los mismos temas pero tienen diferente longitud pueden tener una gran similitud con esta métrica.

La propuesta se basa en aplicar este modelo para tratar a cada oración del texto como un vector n-dimensional (siendo n la cantidad de palabras distintas presentes en el documento), y luego compararlas utilizando la similitud coseno. Las componentes 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.

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

4.3. BM25

Okapi BM25 es una función 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 TF–IDF.

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

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

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 penaliza aquellos términos que aparecen en más de la mitad de los textos, haciendo negativo a su valor. Como este comportamiento no es compatible con el algoritmo de PageRank, se utiliza la siguiente fórmula:

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

donde ε ronda entre 0,5 y 0,30 y $avgIDF$ es el idf promedio para todos los términos. También se ensayaron las alternativas de igualar a cero el resultado en

caso de ser negativo, y de reemplazar la función inversa por una similar, cuyo resultado es no negativo, o estrictamente positivo.

Existe, además, 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.

5. Ejemplo

Se muestra a continuación un documento del conjunto de evaluación de la conferencia DUC (Document Understanding Conference) del año 2002 [7]:

1. Growth Factor Protects Heart Following Attack, Study In Rats Shows
2. By PAUL RECER AP Science Writer WASHINGTON (AP)
3. A natural substance called transforming growth factor beta appears to be able to limit damage to cardiac cells following a heart attack, according to a study published in the journal Science.
4. In a study at the Jefferson Medical College in Philadelphia, a group of laboratory rats induced to have heart attacks suffered 50 percent less cell damage after injections of transforming growth factor beta than did rats that did not receive the TGF beta.
5. "TGF beta is a growth factor that opposes some of the bad guys following a heart attack,"said Dr. Allan Lefer, a professor at Jefferson.
6. Lefer said his research team simulated heart attacks in 24 rats by partially blocking key arteries in their hearts.
7. In 12 of the rats, the researchers injected a placebo.
8. In the other 12, they injected transforming growth factor beta.
9. For those who received the TGF beta, said Lefer, "the damage from the attack was much less severe.
10. There was about 50 percent less injury with TGF beta than without it".
11. The extent of heart cell damage was determined by measuring the amount of creatine kinase in the heart tissue following an attack.
12. Hearts damaged when the blood supply is interrupted, as in a heart attack, tend to lose creatine kinase, said Lefer.
13. Thus, by measuring for the loss of this substance, researchers could determine the amount of heart damage.
14. Lefer said the TGF beta seems to block the action of other substances, such as tumor necrosis factor, that can cause blood vessels to narrow following a heart attack.
15. Narrowed blood vessels carry less oxygen-rich blood to cells and this causes additional injury following a heart attack.
16. TGF beta is normally present in heart cells, but the study published in Science said that it is missing from rat heart cells damaged by a simulated heart attack.
17. Though TGF beta is produced naturally in the body, Lefer said his research team used a substance produced artificially by Genentech, a California biotechnology firm.
18. Lefer said his team is now conducting additional studies with TGF beta and that any experimental treatment of human heart attack victims with the substance is at least a year away.
19. Jefferson Medical College, where the study was done, is part of Thomas Jefferson University in Philadelphia.
20. Science, which published the study, is the journal of the American Association for the Advancement of Science.

Modelando el texto de entrada en el grafo mostrado en la Figura 1, el algoritmo de TextRank produce el siguiente resumen:

> A natural substance called transforming growth factor beta appears to be able to limit damage to cardiac cells following a heart attack, according to a study published in the journal Science. In a study at the Jefferson Medical College in Philadelphia, a group of laboratory rats induced to have heart attacks suffered 50 percent less cell damage after injections of transforming growth factor beta than did rats that did not receive the TGF beta. Lefer said the TGF beta seems to block the action of other substances, such as tumor necrosis factor, that can cause blood vessels to narrow following a heart attack. TGF beta is normally present in heart cells, but the study published in Science said that it is missing from rat heart cells damaged by a simulated heart attack.

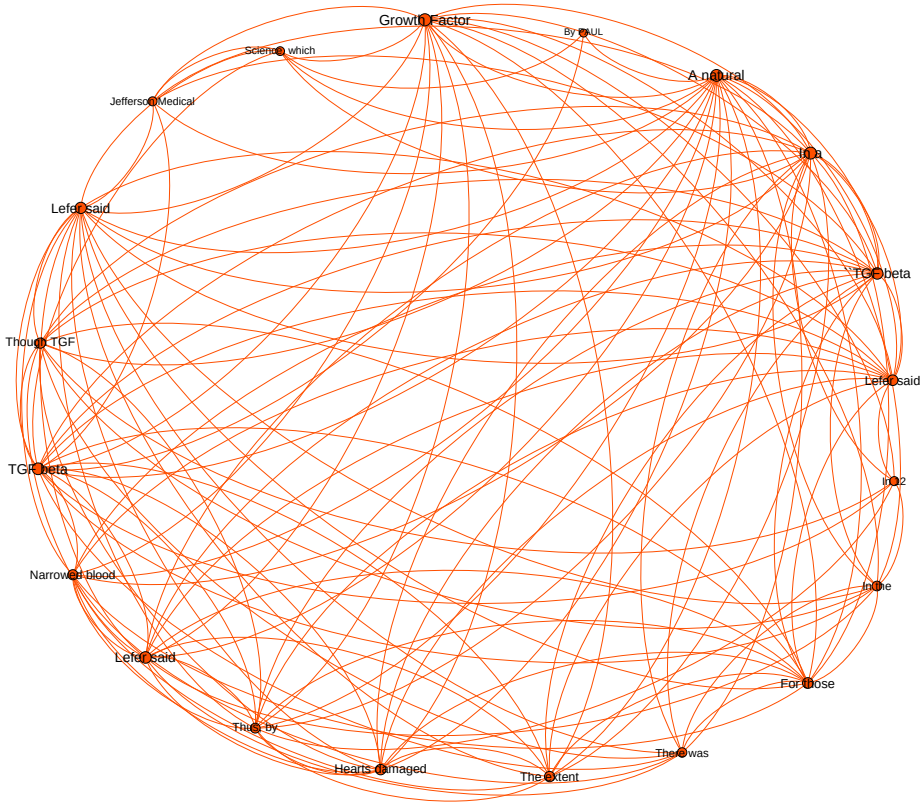

Figura 1. Grafo generado a partir del texto de ejemplo.

6. Evaluación

Para probar el funcionamiento del algoritmo 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 [7]. El corpus cuenta con 567 documentos que fueron resumidos a cerca del 20 % de su tamaño, y es el mismo que el usado en la presentación del algoritmo TextRank en [4].

Para llevar a cabo la evaluación se usó la versión 1.5.5 del paquete de métricas ROUGE [8]. 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 %. Se obtuvo un único puntaje promediando los tres valores.

Para verificar el funcionamiento de todo el conjunto se desarrolló el método de referencia (*baseline*) usado en [4], 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* en un 2,3 % aproximadamente.

6.1. Resultados obtenidos

Los mejores resultados se obtuvieron usando BM25 y BM25+. El incremento más alto se logró al reemplazar los valores negativos por la constante $\varepsilon = 0{,}25$, arrojando una mejora total de 2,92 % para BM25 y 2,60 % para BM25+. En el Cuadro 1 se detallan las alternativas ensayadas.

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 %

Los tiempos de ejecución también se superaron. Se pudieron procesar los 567 documentos de la base de datos de DUC2002 utilizando un 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 2 se muestra una comparación de los puntajes para las métricas de ROUGE-1 y 2.

7. 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 tiene un desempeño de 2,84 % por sobre el estándar de comparación (*baseline*).

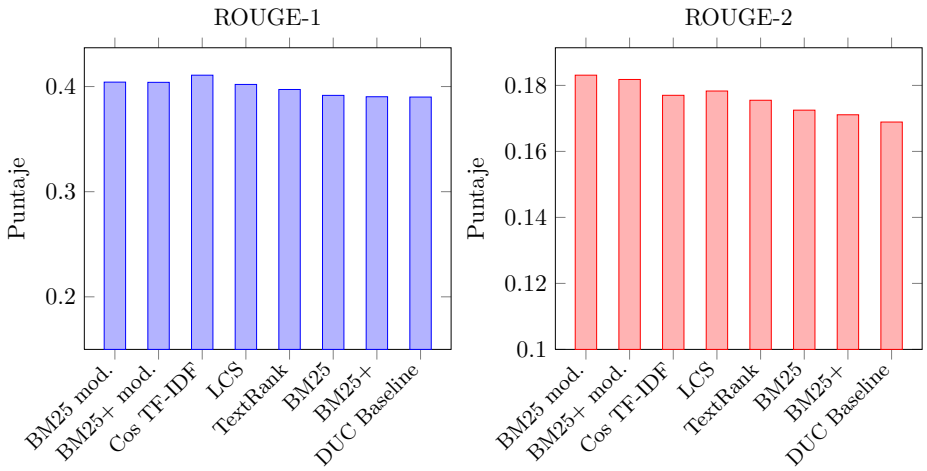

Figura 2. Comparación de puntajes obtenidos para la base de datos de DUC2002 en las métricas ROUGE-1 y 2.

En base a estos resultados proponemos la utilización de BM25 o BM25+ junto con TextRank como herramienta para la generación de resúmenes automáticos. Los resultados obtenidos demuestran mejoras a la versión original del algoritmo con ganancias extra de tiempo de procesamiento.

Queda para próximos trabajos explorar alternativas para la extracción de palabras claves, y la realización de ensayos para explotar aún más el modelo del espacio vectorial.

Referencias

1. Luhn, H. P.: *The Automatic Creation of Literature Abstracts.* IBM J. Res. Dev., 2(2):159–165, Abril 1958, ISSN 0018-8646. `http://dx.doi.org/10.1147/rd.22.0159`.
2. 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`.
3. Salton, G., A. Singhal, M. Mitra y C. Buckley: *Automatic Text Structuring and Summarization.* Information Processing and Management, 33(2):193 – 207, 1997.
4. 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.
5. 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`.
6. 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`.

7. Document Understanding Conference: *DUC 2002 guidelines*, July 2002. `http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html`, visitado el 2015-04-25.
8. 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.