



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERÍA

MÓDULO DE RESÚMENES AUTOMÁTICOS  
BASADO EN TEXTRANK CON INTEGRACIÓN  
A GENSIM

Barrios, Federico – 92954  
López, Federico – 92278

Mayo, 2015



# Índice

<b>1. Motivación</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Introducción teórica</b>	<b>2</b>
3.1. Resúmenes: generación y clasificación . . . . .	2
3.1.1. Enfoque abstractivo y extractivo . . . . .	2
3.1.2. Aprendizaje supervisado y no supervisado . . . . .	3
3.1.3. Fuente de los datos . . . . .	3
3.1.4. Propósito del resumen . . . . .	3
3.2. Extracción de palabras claves . . . . .	3
<b>4. Trabajo previo</b>	<b>3</b>
<b>5. Herramientas y fundamentos</b>	<b>4</b>
5.1. PageRank . . . . .	4
5.2. TextRank . . . . .	4
5.3. Gensim . . . . .	4
5.4. Evaluación de resúmenes . . . . .	5
<b>6. Implementación</b>	<b>5</b>
6.1. Preprocesamiento . . . . .	5
6.1.1. Separación del texto . . . . .	6
6.2. Generación de resúmenes automáticos . . . . .	6
<b>7. Experimentos</b>	<b>6</b>
7.1. Evaluación . . . . .	6
7.2. Propuestas . . . . .	7
7.2.1. Subcadena común . . . . .	7
7.2.2. Similitud coseno . . . . .	7
7.2.3. BM25 . . . . .	7
7.3. Resultados . . . . .	8
<b>8. Conclusiones</b>	<b>8</b>



## 1. Motivación

Los resúmenes automatizados son muy utilizados en tareas relacionadas con el procesamiento de lenguaje natural y de aprendizaje automático. Su uso en motores de búsqueda, por ejemplo, mejora la eficiencia de indexación de textos y a su vez asiste en la presentación de resultados de manera efectiva. El incremento en la cantidad de información disponible en Internet ha intensificado su utilización en los últimos años y, en consecuencia, se ha dedicado enorme esfuerzo para mejorar los algoritmos existentes.

## 2. Objetivos

El presente trabajo consta de tres objetivos principales:

- Desarrollar el módulo para generar resúmenes automáticos usando un algoritmo conocido.
- Analizar, diseñar e implementar modificaciones para intentar mejorar el rendimiento del algoritmo seleccionado.
- Integrar las implementaciones a una herramienta de código abierto de procesamiento del lenguaje natural.

## 3. Introducción teórica

### 3.1. Resúmenes: generación y clasificación

Un resumen es una reducción a términos breves y precisos de lo esencial de una fuente de información. Su objetivo es el de extraer contenido intentando sintetizar sus conceptos más importantes, y su uso es altamente benéfico en tareas de aprendizaje debido a que:

- Facilitan la selección de información,
- Acortan tiempos de lectura,
- Simplifican búsquedas en textos,
- Optimizan la creación de índices.

La investigación indica, además, que contribuyen en tareas automatizadas: su utilización para presentar resultados de motores de búsquedas atrajo el interés de académicos desde principios de la década del 2000, convirtiéndose hoy en día en una funcionalidad básica de los principales buscadores de Internet.

Sin embargo, si bien esta tarea puede no resultar costosa para un ser humano, durante muchos años fue considerada como difícil de automatizar. Este tema es uno de los que investiga el campo de procesamiento de lenguaje natural, dedicado a facilitar la interacción entre las computadoras y los seres humanos.

#### 3.1.1. Enfoque abstractivo y extractivo

Existen dos métodos para generar resúmenes automatizados: abstractivos y extractivos.

Los primeros se construyen regenerando el contenido extraído del texto original; esto es, se reformulan las frases por medio de técnicas de generación de lenguaje natural: fusión, combinación o supresión de términos. De esta manera, se obtienen pasajes que en principio no pertenecían al texto de origen, similar al modo en que lo harían las personas.

Por el contrario, los resúmenes extractivos se crean a partir de la selección de un conjunto de palabras u oraciones consideradas sobresalientes en el texto original. Las oraciones se obtienen literalmente, se unen libremente, y se presentan con el objetivo de crear un resumen del texto.

Las técnicas abstractivas son un área de creciente interés. Sin embargo, debido a la complejidad y las restricciones técnicas aparejadas, la investigación se ha volcado hacia el enfoque extractivo.



### 3.1.2. Aprendizaje supervisado y no supervisado

En el campo del aprendizaje automático, a los algoritmos que requieren de un conjunto de entrenamiento con datos etiquetados se denominan supervisados. Analizando estos ejemplos de entrenamiento, el sistema infiere funciones que pueden ser usadas para aplicarlas a nueva información.

Por el contrario, a los algoritmos que no son entrenados se los denomina no supervisados. Este grupo utiliza técnicas para encontrar patrones o grupos ocultos en los datos a analizar.

### 3.1.3. Fuente de los datos

Los algoritmos de generación de resúmenes también se categorizan de acuerdo a la cantidad de documentos que se toman como fuente, determinando sumarización de un documento y multi-documento.

Para este último caso se proporcionan textos con una temática afín. El resumen resultante permite a los lectores familiarizarse rápidamente con los tópicos principales de una colección de documentos.

### 3.1.4. Propósito del resumen

Los resúmenes de documentos pueden crearse de manera genérica o bien basados en consultas.

Aquéllos resúmenes basados en consultas favorecen temas o aspectos específicos de un texto, de acuerdo a dicha consulta; teniendo en cuenta sus palabras clave o los tópicos a los que hace referencia. Los resúmenes genéricos, por otro lado, proporcionan los temas más relevantes tratados en el texto por sí solo.

## 3.2. Extracción de palabras claves

La extracción de palabras claves consiste en identificar el conjunto de términos que mejor describen a un documento de manera automatizada. Esto sirve como introducción a sus conceptos más relevantes, pues uno de los primeros pasos para modelar el conocimiento de una comunidad es tomar vocabulario del dominio. Es por ello que se han estudiado métodos para extraer este léxico técnico en base a documentos del área o comunidad en estudio.

Dichas palabras se pueden utilizar para construir índices de colecciones, clasificar textos, extraer terminología de dominio específica, o incluso pueden servir como un breve resumen.

El enfoque más sencillo para esta tarea ha empleado un criterio basado en la frecuencia de las palabras. Sin embargo, se han investigado otras debido a los malos resultados de esta técnica. Las principales se basan en métodos supervisados, donde el sistema es entrenado para reconocer palabras claves tomando en cuenta cuestiones sintácticas y/o léxicas.

En la sección 'TextRank para palabras claves' se analiza y desarrolla la aplicación de TextRank como método no supervisado para esta tarea. Los resultados obtenidos se equiparan los de las técnicas principales en el área.

## 4. 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. También, se han desarrollado modelos estadísticos basados en corpus de entrenamiento para combinar varias heurísticas: palabras clave, posición de las oraciones, longitud de las oraciones, frecuencia de palabras y palabras contenidas en 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 las unidades que juegan un papel preponderante dentro de la estructura discursiva [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 & Tarau como por Erkan & Radev [6], consiste en utilizar el prestigio de las unidades léxicas



(oraciones o palabras) dentro del grafo. Dicha técnica ha sido la adoptada en el presente trabajo.

## 5. Herramientas y fundamentos

### 5.1. PageRank

PageRank es un algoritmo creado por Larry Page y Sergey Brin, fundadores de Google, en 1996, como parte de una investigación sobre motores de búsqueda. Su objetivo es asignar un valor numérico a cada nodo en una red, con el propósito de medir la importancia relativa de cada elemento del conjunto. Para esto analiza los vínculos que componen la red. La idea subyacente es que los nodos más relevantes de la red serán los que tengan la mayor cantidad y calidad de conexiones con otros nodos.

En sus orígenes, PageRank fue pensado para aplicarse sobre la red de Internet (World Wide Web), pero puede aplicarse a cualquier otra red para estimar sus elementos principales. En términos matemáticos, PageRank brinda una manera iterativa de calcular el autovector principal de la matriz de adyacencia del grafo. Esto da como resultado la distribución de probabilidad de acceder a los nodos de la red, navegando de manera aleatoria a través de los vínculos o aristas.

Una de las características más notables del algoritmo es que toma en cuenta la información global de toda la red, y no sólo lo concerniente a cada nodo en particular. Si se arma un grafo en base a un documento, con las unidades léxicas o semánticas como vértices, se puede aplicar PageRank para priorizar los nodos principales. De esta manera, se pueden obtener las oraciones o palabras principales, utilizando información proveniente de todo el texto. En este enfoque está basado TextRank.

### 5.2. TextRank

TextRank es un algoritmo no supervisado basado en grafos para realizar resúmenes automáticos extractivos y/u obtener palabras claves de un texto. Fue presentado en 2004 por Rada Mihalcea y Paul Tarau en el paper “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. Por este motivo, TextRank puede ser aplicado a cualquier texto, incluso en distintos idiomas, generando un resumen basado sólo en las propiedades intrínsecas del texto.

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.

Las particularidades de los casos de creación de resúmenes y extracción de palabras claves se analizan en la sección ‘Implementación’.

### 5.3. Gensim

Es una biblioteca de código libre, escrita en Python para el modelaje de tópicos y la indexación de documentos que está diseñada para trabajar con conjuntos de textos de gran tamaño. Su uso se ha expandido tanto en lo comercial como lo académico, apuntando especialmente al área de procesamiento del lenguaje natural y a la búsqueda y recuperación de la información.



Gensim utiliza un paradigma muy poderoso en el área de procesamiento de lenguaje natural, denominado VSM (modelo de espacio de vectores), en el cual se representan los documentos de un corpus como vectores de un espacio de muchas dimensiones. Esta representación explota la idea de que los textos en el área del modelado de tópicos se pueden expresar según un número de conceptos, lo que aumenta la eficiencia y también ayuda a eliminar ruido.

La biblioteca está diseñada para proveer independencia del tamaño del corpus, es decir, para poder analizar textos que sean más grandes que la memoria RAM del sistema. El proyecto, además apunta a exponer una interfaz mínima e intuitiva utilizando terminología de procesamiento de lenguaje.

Gensim proporciona implementaciones de algoritmos como TF-IDF, análisis semántico latente, proyecciones aleatorias y alocaión de Dirichlet latente.

## 5.4. Evaluación de resúmenes

La herramienta por defecto para evaluar los sistemas generadores de resúmenes es Rouge (Recall-Oriented Understudy for Gisting Evaluation). Este método fue desarrollado en la universidad de Southern California para automatizar la comparación entre resúmenes generados por el sistema bajo evaluación contra otros (ideales) escritos por seres humanos, tomados como referencia.

Históricamente, los puntajes asignados en tareas de competencias como DUC fueron manuales; juzgando coherencia, capitalización, cohesión, consistencia y contenido [9]. Todo este trabajo es caro y difícil de coordinar, demandando de al menos 3.000 horas hombre. El surgimiento de Rouge atiende la necesidad de automatizar ese proceso, reemplazando un método llamado BLEU, cuyos resultados no siempre se correlacionaban con los asignados por los jurados.

El mecanismo de funcionamiento de los métodos del paquete Rouge consiste en calcular la sensibilidad (recall) de unidades léxicas entre los resúmenes de sistema y los de referencia. Se generan así diferentes métricas:

- Rouge-N: cuando se tienen en cuenta n-gramas (n palabras). Este método favorece a documentos con palabras compartidas por más candidatos.
- Rouge-L: cuando se tiene en cuenta la subsecuencia máxima común (LCS, longest common subsequence). Este método propone que un conjunto de documentos va a ser más parecido mientras más larga sea la subsecuencia común máxima.
- Rouge-W: le asigna pesos a las subsecuencias según su largo.
- Rouge-S: es similar a Rouge-2, teniendo en cuenta bigramas pero permitiendo una cantidad arbitraria de espacios entre palabras.
- Rouge-SU: extiende la idea de Rouge-S, pero además le otorga valor a las palabras que haya en común, sin importar que no verifiquen un orden.

Rouge asigna un puntaje entre cero y uno a un esquema de evaluación conformado por una serie de documentos generados y sus referencias. Por diseño, se espera que buenos resúmenes tengan puntajes más altos; pero inevitablemente se requiere de trabajo humano para proveer las referencias.

Para verificar el funcionamiento de Rouge se han usado las bases de datos de las conferencias DUC de los años 2001, 2002 y 2003. Se compararon las evaluaciones con los puntajes otorgados manualmente por los jurados para encontrar que todas las métricas arrojan muy buenos resultados; y a partir de la edición 2004 las evaluaciones se llevan a cabo usando exclusivamente Rouge.

## 6. Implementación

### 6.1. Preprocesamiento

El preprocesamiento del texto es una parte esencial en cualquier sistema que trate con datos del lenguaje natural. En esta fase se identifican las unidades léxicas que serán analizadas por la aplicación en las etapas subsiguientes.

La calidad de este proceso tendrá un gran impacto en los resultados obtenidos. Es por ello que se deben seleccionar unidades léxicas significativas. Es decir, aquellas que contengan las características lingüísticas relevantes para el caso en estudio. A su vez, se deben filtrar las que no aportan información útil. En este proceso también, se unifica la codificación que pudiera existir en los distintos textos.



Aquí se detallan las etapas principales.

### 6.1.1. Separación del texto

Esta etapa, también conocida como tokenización consiste en separar el texto en oraciones o palabras. Separar palabras suele resultar más sencillo, dado que los espacios en blanco son una buena aproximación como límite de separación. Se debe tener un tratamiento especial de los casos tales como “can’t” o “don’t” en idiomas como el inglés.

Separar por oraciones suele ser caso más complejo. Definir los límites es una cuestión realmente problemática en el área del lenguaje natural. Generalmente se utilizan los símbolos .!? como separadores, pero se deben tratar con cuidado excepciones como *Ph.D.*, *Dr.*, *Lic.*, *\$19.99*, *p.m.* o *RR.HH.*

El enfoque más utilizado para resolver esta cuestión se basa en la aplicación de expresiones regulares teniendo en cuenta reglas como las siguientes:

1. El punto (“.”) finaliza una oración si la siguiente palabra comienza con mayúscula.
2. Si la palabra precedente al punto pertenece a un listado de abreviaturas preestablecido, entonces no termina la oración.

Otras alternativas consisten en el uso de algoritmos entrenados con datos específicos del área sobre la cual se esté trabajando.

## 6.2. 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 [10].

Inicialmente, se construye un grafo en base al texto. En este caso, se agrega cada oración como vértice. Para crear las aristas con su peso correspondiente, se debe definir una función de similitud entre dos oraciones dadas. Esta función de similitud será la que dicte cuánto una oración “recomienda” a otra, dado que ambas poseen contenidos similares y abordan los mismos conceptos.

La función utilizada formalmente se define de la siguiente manera:

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

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

El resultado de este proceso es el texto representado como un grafo ponderado y altamente conexo. En base a esto se aplica PageRank para priorizar los vértices más relevantes.

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

## 7. Experimentos

### 7.1. 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. 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 [11]. 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 el paper, que construye el resumen extrayendo las primeras oraciones de cada artículo. Se comprobó que los resultados fueron los mismos que en el paper: la versión original de TextRank mejora al baseline por cerca de un 2,3 %.



## 7.2. Propuestas

### 7.2.1. 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.

### 7.2.2. 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 debe modelar a los documentos como vectores. Para ello se utiliza el modelo del espacio vectorial.

El modelo del espacio vectorial es un modelo algebraico 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 el modelo del espacio vectorial 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.

### 7.2.3. 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.

Dadas dos oraciones R,S, la misma se define como:

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

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





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 %
<b>TextRank</b>	<b>0.3983</b>	<b>0.1762</b>	<b>0.1948</b>	-
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 %

Existe una variante de este método, conocida como BM25+. Esta repara deficiencias de BM25 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.

### 7.3. Resultados

Los resultados óptimos 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 obtenían 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, alcanzado así resultados sobresalientes. En la Tabla 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.

GRAFICO!!!!

## 8. 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.

Las evaluaciones fueron hechas con las mismas técnicas y datasets 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](http://citeseer.nj.nec.com/page98pagerank.html).
- [9] *Procedure for human comparison of model (reference) and peer (system-generated and other) abstracts*. <http://www-nlpir.nist.gov/projects/duc/duc2002/protocol.html>, 2002.
- [10] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [11] 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.