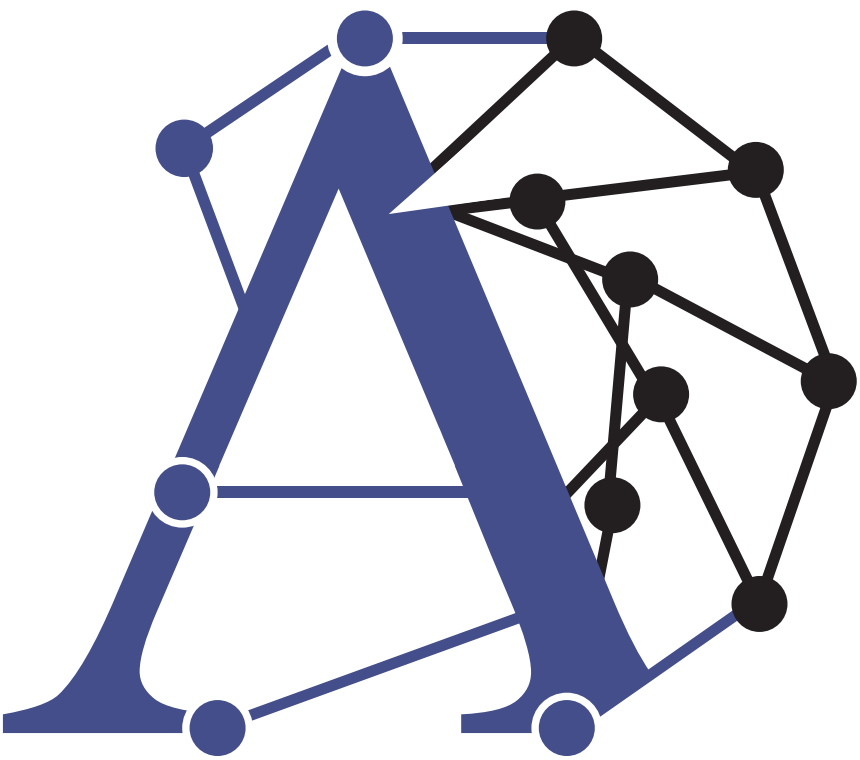
UAEM | Universidad Autónoma del Estado de México

Generación automática de resúmenes

Retos, propuestas y experimentos

Automatic Generation of Text Summaries

Challenges, proposals and experiments

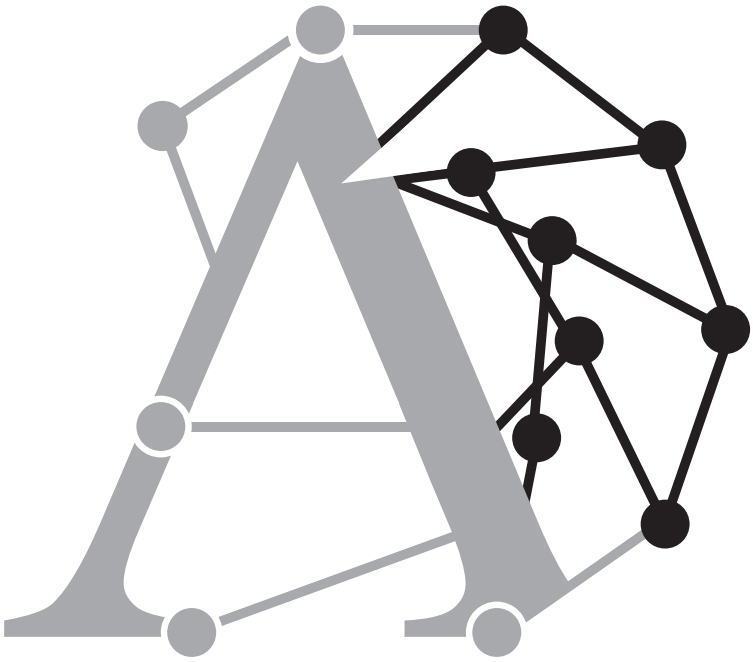


Yulia Nikolaevna **Ledeneva**
René Arnulfo **García Hernández**

investigación universitaria

Generación automática de resúmenes

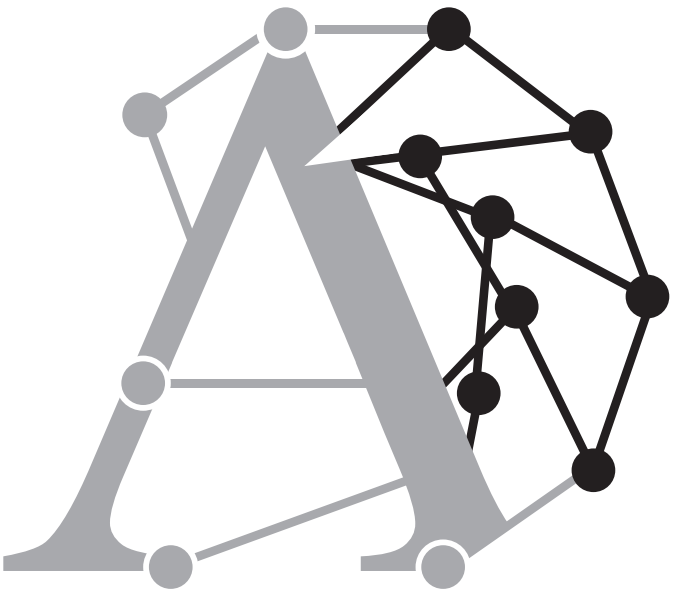
Retos, propuestas y experimentos



UAEM | Universidad Autónoma del Estado de México

Generación automática de resúmenes
Retos, propuestas y experimentos

Yulia Nikolaevna Ledeneva
René Arnulfo García Hernández



UAEM | Universidad Autónoma del Estado de México

Generación automática de resúmenes
Retos, propuestas y experimentos

Automatic Generation of Text Summaries
Challenges, proposals and experiments

Traducción del español a inglés: Luis Cejudo Espinosa

Libro de investigación arbitrado por pares ciegos, con base en los criterios establecidos por la Secretaría de Investigación y Estudios Avanzados.

1a edición, enero 2017

ISBN: 978-607-422-782-6

Instituto Literario núm. 100 Ote., Centro, C.P. 50000, Toluca, México
http://www.uaemex.mx

Impreso y hecho en México
Printed and made in Mexico

A mis hijas
Renata Yulie
María Constanza

Contenido

Lista de tablas

Lista de figuras

INTRODUCCIÓN

En las últimas dos décadas, el aumento exponencial de información electrónica ha vuelto necesario comprender, de manera rápida, el contenido esencial de grandes volúmenes de información textual. El primer paso es discernir aquella información que es de nuestro interés. De hecho, se estima que el 80 % de la información electrónica de una empresa se encuentra en forma de texto y el otro 20 % en bases de datos. Este porcentaje aumenta cuando se habla de Internet, pues la mayoría de la información se encuentra en forma textual, es decir, en forma de lenguaje natural.

Esto señala la importancia del desarrollo de métodos automáticos que permitan detectar el contenido más relevante de un documento, con el fin de producir un texto más corto a manera de resumen. La generación automática de resúmenes de texto (GART) es una de las tareas prioritarias del área de investigación de procesamiento de lenguaje natural, la cual busca producir resúmenes similares a los creados por humanos. Para ello, se han generado algunos modelos y métodos imprácticos, desde nuestro punto de vista, ya que no pueden trabajar de manera independiente del lenguaje, del dominio del documento de texto o deben ser alimentados con una serie de recursos y procesos lingüísticos (diccionarios, taxonomías, gramáticas, analizadores sintácticos, etc.), que requieren una alta intervención del trabajo humano. No por ello estas propuestas dejan de ser interesantes o prometedoras en un futuro. En este sentido, y con el fin de tener un estado del arte más completo sobre esta área de investigación, no sólo se presenta una revisión de los métodos desarrollados para este propósito, sino una evaluación de las herramientas comerciales de GART.

Este libro presenta un método computacional novedoso, a nivel internacional, para la generación automática de resúmenes de texto, pues supera la calidad de los que actualmente se pueden crear. Es decir, es resultado de una investigación que buscó métodos y modelos computacionales lo menos dependientes del lenguaje y dominio.

Una de las aportaciones derivadas de la investigación fue el diseñó de una metodología que permite saber cuál sería la calidad máxima que un método de GART puede obtener de una colección de documentos determinada. Con este parámetro, desconocido hasta ahora, es posible conocer qué tan significativo es el avance que están presentando los nuevos métodos y herramientas comerciales de GART, y con ello se puede saber qué tanto le queda por descubrir a esta área de investigación.

Otra de las aportaciones de este libro se encuentra en afirmar que todos los métodos de GART presentan cuatro etapas en sus procesos: selección de términos, pesado o ponderación de términos, pesado o ponderación de oraciones y selección de oraciones para el resumen. Esto permite que se puedan analizar los métodos ya propuestos, pero que además se puedan generar nuevas y mejores investigaciones si se presentan innovaciones en cada una de estas etapas. Recordemos el principio axiomático de resolución de problemas, "divide y vencerás". En este sentido, en este libro se presentan diversas innovaciones para cada una de las etapas de la GART y las experimentaciones correspondientes que validan el método propuesto.

En específico, para la primera etapa, selección de términos, se describe cómo detectar descripciones multipalabras que conlleven un significado importante. Para ello, proponemos utilizar las secuencias de palabras que frecuentemente se repiten en el texto original, pretendiendo que a mayor frecuencia sea posible caracterizar de qué trata el texto. Sin embargo, como pueden haber muchas secuencias frecuentes (SFs) se propone sólo utilizar las secuencias frecuentes maximales (SFMs), las cuales no son subsecuencia de alguna otra secuencia frecuente. Al utilizar las SFMs se pretende enriquecer el significado de cada término, puesto que una SFM representa a todo un conjunto de secuencias frecuentes. De hecho, a partir del conjunto de SFMs es posible obtener todas las secuencias frecuentes.

También para las etapas de pesado de términos, pesado de oraciones y selección de oraciones se presentan innovaciones que nos permitieron desarrollar un método computacional para al problema de la GART. El nuevo método desarrollado ha obtenido resultados superiores al estado de arte considerando: colecciones de noticias estándar que se utilizan para probar los nuevos métodos propuestos a nivel internacional.

Los estudiantes e investigadores en el área de procesamiento de lenguaje natural, inteligencia artificial, ciencias computacionales y lingüística computacional serán quizá los primeros interesados en este libro. No obstante, también se pretende introducir a público no especializado en esta prometedora área de investigación; por ello, hemos traducido al español algunos tecnicismos y anglicismos, propios de esta disciplina, pero sin dejar de mencionar, en todo momento, su término en inglés para evitar confusiones y lograr que aquellos lectores interesados puedan ampliar sus fuentes de conocimiento.

CAPÍTULO I.
LOS PLANTEAMIENTOS BÁSICOS

En este capítulo se presentan algunos conceptos básicos para comprender el problema de investigación que se aborda en este libro. Posteriormente, se exponen los objetivos que se persiguen con el desarrollo del nuevo método para generar resúmenes automáticos de textos, así como la metodología para conseguirlos; también se describe lo que podemos aprender en este libro. Por último, se incluye la organización de esta obra.

I.1 Introducción

En las últimas dos décadas, el aumento exponencial de la información electrónica ha vuelto necesario comprender, de manera rápida, el contenido esencial de grandes volúmenes de información textual. El primer paso es discernir qué información, de toda la disponible, es de nuestro interés. De hecho, se estima que el 80 % de la información electrónica de una empresa se encuentra en forma de texto y el otro 20 % en bases de datos (Leavitt, 2002). Este porcentaje se queda corto cuando se habla de Internet, pues la mayoría de la información se encuentra en forma textual, es decir, en forma de lenguaje natural.

El resumen de un documento escrito produce un texto bastante más corto que el original, pero sin omitir la información más relevante del mismo. Hay una serie de escenarios donde la construcción automática de tales resúmenes resulta de utilidad. Por ejemplo, un sistema de recuperación de información podría presentar los resúmenes generados automáticamente de una determinada lista de documentos, así el usuario podría decidir rápidamente cuáles son de su interés y debe consultar a detalle. Esto es lo que en cierto grado Google modela con los fragmentos mostrados en sus resultados de búsqueda. En la figura I.1 se pueden ver los títulos o partes de los fragmentos de texto recuperados por Google al introducir la frase "generación de resumen". De los primeros cuatro documentos recuperados, para el presente trabajo sólo interesaría el primero

de ellos. Sin embargo, habría que revisar otros 34,296 documentos donde podría encontrarse la información buscada.

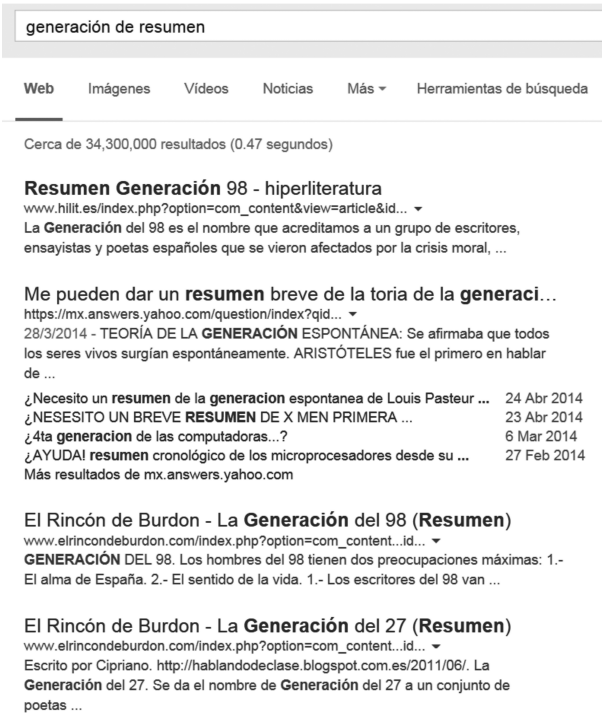


Figura I.1 Primeros cuatro documentos recuperados en Google con la consulta "generación de resumen", realizada el 8 mayo del 2014.

Otros ejemplos incluyen la construcción automática de resúmenes de artículos periodísticos, ya que las propias agencias noticiosas pueden manejar cientos de noticias al día en diversas categorías (Evans & McKeown, 2005; McKeown, Barzilay, Chen, Elson, Evans, *et al.*, 2003; Nenkova, Siddharthan & McKeown, 2005). Por ejemplo, la agencia de noticias del Estado mexicano, Notimex, maneja alrededor de 500 noticias por día. En este sentido, las empresas y organizaciones reciben varios mensajes de correo electrónico que podrían resumirse y ser enviados a los dispositivos móviles como mensajes SMS (Corston-Oliver, Ringger, Gamon & Campbell, 2004; Shrestha & McKeown, 2004; Wan & McKeown, 2004). Actualmente, los partidos políticos, los candidatos presidenciales y las empresas especializadas en lanzamientos de nuevos productos, como películas o canciones, se sienten obligados a saber, en resumen y de forma inmediata, qué se está opinando sobre ellos en las redes sociales. Sin

embargo, en redes como Twitter y Facebook se publican miles de mensajes por segundo, ante diversos eventos, lo que hace imposible al ojo humano leerlos en corto tiempo. Por ello, se hace necesario contar con herramientas generadoras de resúmenes que, junto con herramientas de minería de opinión, puedan satisfacer dichas necesidades.

Esta necesidad de resumir la información también la tienen los investigadores de diferentes áreas, pues diariamente se publican nuevos avances tecnológicos en revistas y textos especializados; en otras disciplinas, como las jurídicas, la importancia de contar con resúmenes es vital al revisar la gran cantidad de información disponible, en textos legales, por ejemplo (Farzindar & Lapalme, 2004). En nuestros días, los teléfonos inteligentes y tabletas permiten disponer de grandes cantidades de información textual, ya sea que ésta se haya descargado previamente o se consulte al momento. Sin embargo, las pantallas de estos dispositivos pueden ser bastante pequeñas, por lo que se hace cansado leer todo el texto del documento. Una herramienta de GART en línea o instalada en el propio dispositivo puede solucionar este problema (Futrelle, 2004; Dia & Shan, 2006; Otterbacher, Radev & Kareem, 2006; Carberry, Elzer, Green, McCoy & Chester, 2004; Carberry, Elzer & Demir, 2006).

Como se puede apreciar, las necesidades de la GART son diversas de acuerdo a sus contextos, dispositivos y usuarios. Por ello, en esta investigación se considera indispensable el desarrollo de métodos de resúmenes de tipo genérico, que dependan lo menos posible del dominio y del lenguaje del texto de entrada, dejándole al usuario el control de la longitud del resumen de salida.

Hoy en día existen varias herramientas de GART, gratuitas o comerciales, que permiten generar resúmenes. Entre ellas se encuentran Copernic Summarizer (Copernic, 2013), Microsoft Office, Svhoong Summarizer (Svhoong, 2013), Pertinence Summarizer (Pertinence, 2010), Tools4noobs Summarizer (Tool4noobs, 2013) y Open Text Summarizer (OTS, 2013). Sin embargo, algunas de ellas sólo pueden trabajar en determinados lenguajes. Parte de este libro se enfocará en conocer las características de dichas herramientas y en evaluar la calidad de los resúmenes que generan. Es decir, será posible conocer cuál es el avance que tienen las herramientas de GART respecto a los métodos reportados en el estado del arte.

La GART es una tarea de investigación del área de procesamiento de lenguaje natural (PLN), que a su vez forma parte de un campo multidisciplinario donde convergen lingüística, matemática, estadística, computación, reconocimiento de patrones, minería de datos, inteligencia artificial y otras disciplinas. Todas estas materias hacen posible que las computadoras entiendan el lenguaje natural escrito. Si esto último fuera posible las computadoras podrían entender desde palabras aisladas,

sonidos y frases, hasta las ideas que contiene el lenguaje; sin duda, esto podría ayudar al ser humano a realizar diversas tareas relacionadas con el lenguaje natural. Las tareas del PLN diferencian muy bien los problemas de lenguaje natural hablado y escrito. A este último campo se le ha denominado como tratamiento automático de texto.

En el tratamiento automático de texto se pueden trabajar aplicaciones simples o complejas. Por ejemplo, separar en guiones una palabra, detectar y corregir errores ortográficos o gramaticales, correctores de hechos y coherencia, visualización y exploración de grandes colecciones de documentos, sistemas de recuperación de información, sistemas de extracción de información, búsqueda de respuestas, traducción automática, detección de plagio, reconocimiento de la autoría de un texto, generación de resúmenes, entre otras.

Los métodos de generación de resúmenes pueden ser clasificados, de acuerdo a su tipo de entrada, en un solo documento y en múltiples documentos. En la GART de un solo documento (en inglés, *Single Automatic Text Summarization*), sólo se construye el resumen de un texto, mientras que en la GART de múltiples documentos (en inglés, *Multidocument Automatic Text Summarization*) se construye el resumen a partir de toda una colección de documentos de entrada (así como todas las noticias del día de hoy o los resultados de la búsqueda para una consulta). En este libro presentamos y experimentamos la GART con un solo documento de entrada.

Los métodos de GART pueden ser clasificados, de acuerdo al tipo de resumen de salida, en resúmenes abstractivos y extractivos (Lin, Hovy, 1997). Un resumen abstractivo es un texto arbitrario que describe el contexto del documento original. El proceso consiste en "comprender" el texto original y "reescribirlo" en menos palabras. Este método utiliza una gran cantidad de sofisticados métodos y recursos lingüísticos para examinar e interpretar el texto en búsqueda de nuevos conceptos y expresiones que permitan reescribir el texto de manera más corta, pero sin perder la información más importante del documento original. Esta puede parecer la mejor manera de construir un resumen, ya que así es como los seres humanos lo hacen, sin embargo, en la vida real la tecnología lingüística para el análisis y generación de texto aún no ha logrado que estos métodos sean viables en la práctica. Este reto aumenta cuando se tiene que considerar los diferentes dominios y lenguajes del texto de entrada.

En este sentido, los seres humanos utilizan su conocimiento de fondo para generar un resumen donde resalte aquello que les parece importante, por ello, cada persona genera resúmenes con diferentes palabras, pero manteniendo, casi siempre, la misma información. Esto mismo

sucede con los métodos de GART abstractivos, aunque pueden llegar a interpretaciones erróneas que alteren el sentido del texto original, pues dependen de la calidad de las fuentes de información, a partir de las cuales se buscan los nuevos conceptos o expresiones.

Por otra parte, un resumen extractivo consiste básicamente en seleccionar oraciones importantes (frases, párrafos, etc.) del texto original y presentarlas al usuario en el mismo orden, es decir, una copia del texto fuente sin las oraciones omitidas. Un método de generación de resúmenes extractivos sólo decide si cada una de las oraciones del texto se incluirá o no en el resumen. El resultado es un sumario o compendio de ideas importantes que podría tener una lectura no fluida. Sin embargo, la sencillez de las técnicas estadísticas subyacentes hace de este resumen una alternativa atractiva y robusta, que además puede obtenerse independientemente del lenguaje gracias a los métodos extractivos más "inteligentes".

Debido al potencial que tienen los métodos de GART, frente a la enorme información textual contenida en los medios electrónicos, en este libro se trata el problema de la GART extractivo a partir de un solo documento; se utilizan lo menos posible recursos y métodos lingüísticos sofisticados, para tratar de ser independientes del lenguaje y del dominio del texto de entrada. Esto no impide que las ideas presentadas aquí puedan ser adaptadas posteriormente para trabajar resúmenes abstractivos o con entrada de múltiples documentos. Es más, a partir del material presentado se podrían utilizar recursos lingüísticos, dependientes del lenguaje y del dominio, para incrementar la calidad del resumen de salida, además sería posible considerar otros requerimientos por parte del usuario.

Para ejemplificar el problema se plantea un ejercicio (realmente interesante si se lleva a cabo):

Considerando las cinco oraciones siguientes[1] de un documento, ¿cuáles serían las dos oraciones que escogería a manera de resumen?

A. El gobierno de Egipto protege las pirámides
B. Las pirámides de Egipto son un patrimonio cultural
C. Las pirámides fueron construidas por los faraones
D. Las pirámides de Egipto fueron tumbas para los faraones de Egipto
E. Un buen gobierno protege su patrimonio cultural

De antemano, cualquier selección que haya realizado es la correcta, pero como se verá más adelante, el criterio entre los humanos sobre

[1] Originally in Spanish. TN.

esta pequeña colección también difiere un poco. Otro de los aspectos importantes a considerar es que usted está empleando todo su conocimiento del idioma español para hacer una interpretación. En específico, fue necesario emplear conocimiento léxico, referente al significado de las palabras; conocimiento sintáctico, referente a las estructuras gramaticales y conocimiento semántico, referente a la interpretación del mensaje en su conjunto. Sin embargo, las diferentes culturas y vivencias de una persona hacen que en cada etapa se puedan tener interpretaciones diferentes, lo cual genera ambigüedad.

En este caso, el método que proponemos sugiere, sin tener conocimiento de fondo o sobre el lenguaje, que usted probablemente seleccionó las oraciones D y B, en ese orden de importancia; luego pudo ser C y E, y la menos probable de aparecer es A. Este ejemplo se irá desarrollando en el libro para ejemplificar cómo se llegó a proponer un método con estas características.

Un método típico de generación de resúmenes extractivos consta de varios pasos, en cada uno de ellos hay diferentes opciones que pueden ser elegidas. Vamos a suponer que las unidades de selección son las oraciones (no obstante pueden ser, frases o párrafos). De esta manera, el objetivo final del proceso de generación de resúmenes extractivos será la selección de oraciones. Una manera adecuada para lograrlo es asignar alguna medida numérica que valore la utilidad de una oración en el resumen, después simplemente se elegirán las mejores. En este orden, el proceso de asignación de pesos de utilidad se llama pesado o ponderación. Una de las maneras de estimar la utilidad de una oración es sumar los pesos de los términos que la componen (este proceso es el pesado o ponderación de términos). Para lograrlo, uno debe decidir cuáles serán los términos individuales, esta tarea es la selección de términos. Los diferentes métodos de generación de resúmenes extractivos pueden ser caracterizados de acuerdo por cómo realizan estas tareas.

En este libro se presenta una nueva forma de seleccionar y pesar tanto los términos como las oraciones. Además, se analizan varias opciones simples para la selección estadística de términos con base en unidades más grandes que la palabra. Es decir, se prueban nuevos términos, llamados descripciones multipalabras, que prometen ser una buena selección al generar automáticamente resúmenes de texto.

La originalidad de esta obra consiste en presentar un método computacional nuevo, basado en la estructura propia del texto y con poca dependencia del lenguaje, para mejorar la GART. Este método ha obtenido resultados superiores a los que pueden generar las herramientas comerciales y los métodos del estado de arte. Esto se logró utilizando

colecciones de noticias estándar, creadas específicamente para probar los nuevos métodos propuestos a nivel internacional.

I.2 Objetivos del libro

1. Identificar los mejores métodos y herramientas comerciales que se han desarrollado para el problema de la GART
2. Identificar las etapas básicas que todos los métodos de GART siguen en su trabajo
3. Identificar los recursos lingüísticos y metodológicos que se utilizan en las investigaciones del estado del arte en los modelos de texto y métodos de GART
4. Identificar los corpus de documentos y los sistemas de evaluación comúnmente empleados para probar los nuevos métodos propuestos en la GART
5. Conocer la calidad de los resúmenes generados por las herramientas comerciales de GART. Por lo tanto, se evaluarán con colecciones de noticias estándar
6. Conocer cuál es la máxima calidad posible que puede alcanzar un método de GART para la colección de prueba utilizada
7. Buscar modelos de texto que permitan seleccionar términos con significados enriquecidos semánticamente, basados en el descubrimiento de descripciones multipalabras
8. Desarrollar métodos de GART con calidad superior a los métodos del estado del arte. Con ello se espera que el sistema sea útil para los usuarios y que también sirva como un marco de trabajo
9. Buscar que los métodos a desarrollar para la GART dependan lo menos posible del lenguaje y del dominio

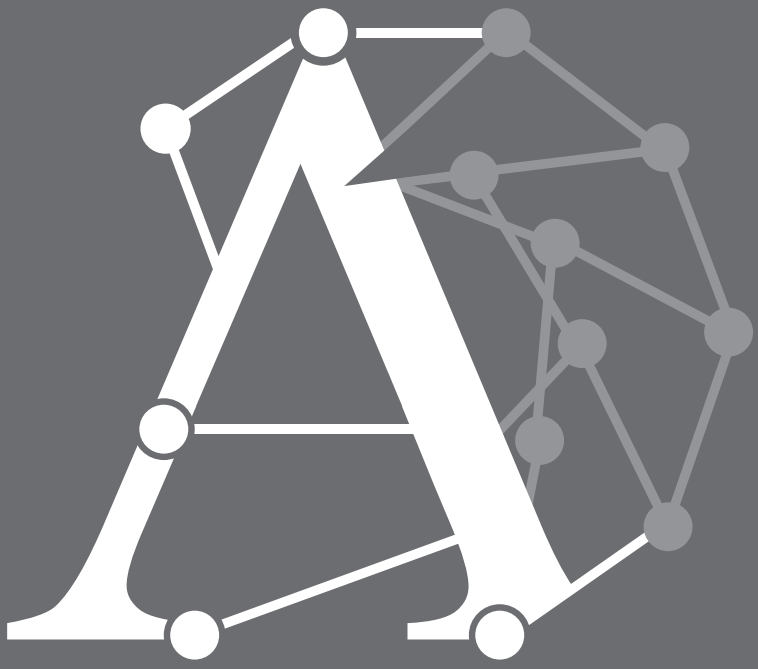
I.3 Lo que podemos aprender de este libro

- Identificar los pasos generales que sigue un método de generación automática de resúmenes de texto extractivo.
- Una introducción al área del procesamiento del lenguaje natural desde la tarea de investigación de la GART.
- Las asociaciones, instituciones, laboratorios, congresos e investigadores de México que están trabajando en el área de PLN.

- Descripción de nuevos métodos para la generación de resúmenes de texto basados en la extracción de nuevos términos multipalabra.
- Desarrollo de nuevos métodos para la generación de resúmenes de texto a partir de un único documento e independiente del lenguaje y del domino.
- Nuevos métodos para la generación automática de resúmenes de texto con resultados superiores a los métodos del estado del arte.
- Identificar los recursos lingüísticos y metodológicos que se utilizan en las investigaciones del estado del arte en los modelos de texto y métodos de GART.
- Identificar los corpus de documentos y los sistemas de evaluación comúnmente empleados para probar los métodos de GART.

I.4 Organización del libro

Este libro está organizado en seis capítulos. En este primer capítulo se introduce el problema de investigación, los objetivos planteados y lo que se espera del libro. El siguiente capítulo resume los métodos del estado del arte. En el tercer capítulo se describe el marco teórico. En el cuarto capítulo se presenta la propuesta de un método para la generación automática de resúmenes de texto, a partir de un solo documento. En el quinto capítulo se presentan los resultados experimentales. Finalmente, en el último capítulo se presentan las conclusiones.



CAPÍTULO II.
MÉTODOS DEL ESTADO DEL ARTE

Este capítulo está dedicado a la presentación detallada del estado del arte. En la sección II.1 se presentará, brevemente, el área de procesamiento de lenguaje natural (PLN) y sus aplicaciones. En la subsección II.1.1 se mostrará el estado que guarda el área del PLN en México, para ello se identifican asociaciones, laboratorios, institutos, investigadores y los congresos principales de dicha área. A continuación, en el apartado II.2, se presenta una introducción del estado del arte de la generación de resúmenes. En específico, la subsección II.2.1 presenta una descripción detallada del estado del arte de la generación de resúmenes extractivos. Esta sección está ordenada tomando en cuenta los cuatro pasos que sigue un método de generación automática de resúmenes extractivos. En el punto II.2.2 se ofrece una descripción breve del estado del arte sobre la generación automática de resúmenes abstractivos. En la sección II.3 se muestran las medidas de evaluación que comparan la calidad de los resúmenes generados automáticamente con los elaborados por humanos. Por último, en la II.4 se muestra la evaluación que se realizó a siete herramientas comerciales y tres métodos del estado del arte de GART para la colección de documentos DUC-2002.

II.1 Procesamiento de lenguaje natural

El área de procesamiento de lenguaje natural (PLN) tiene sus inicios en la inteligencia artificial, cuando se quería dotar a las computadoras de un sistema de inteligencia capaz de procesar el lenguaje natural para entender, reproducir, inferir y deducir el conocimiento presente en la información. En sus inicios, los primeros modelos de PLN venían, principalmente, de la matemática, la estadística, la computación y la lingüística. De ahí que haya surgido el área de lingüística computacional (LC), en la cual se describe cómo funcionan los sistemas de PLN, cómo compilar los datos para la necesidad que tienen estos sistemas; además se resuelve la desambiguación del sentido de las palabras, se construyen diccionarios y bases de datos, se recupera información, se traduce de forma automática de un idioma a otro, etc. (Bolshakov & Gelbukh, 2004a). Como muchas otras áreas (por ejemplo, como las áreas de mecánica y química), la LC tiene la necesidad del procesamiento inteligente de herramientas y la automatización de las tareas de PLN.

Las tareas del PLN diferencian muy bien cuando se tratan problemas de lenguaje natural hablado o escrito. Este último es denominado como tratamiento automático de texto.

En el tratamiento automático de texto se pueden trabajar aplicaciones simples y complejas. Por ejemplo, separar en guiones una palabra, detectar y corregir errores ortográficos o gramaticales, elaborar correctores de hechos y coherencia, sistemas de visualización y exploración de

grandes colecciones de documentos, realizar sistemas de recuperación de información, sistemas de extracción de información, búsqueda de una respuesta a una pregunta realizada, traducción automática, resolver la detección de plagio y reconocimiento de la autoría de un texto, así como la generación de resúmenes, entre muchas otras.

A continuación describimos, con mayor detalle, algunas de las tareas del tratamiento automático de texto:

- Desambiguación del sentido de la palabra (DSP) (Gelbukh & Sidorov, 2002; Manning & Schütze, 1999). Resuelve qué sentido o significado tiene una palabra dada, generalmente en función de su contexto. Esta tarea es muy importante, ya que de su éxito depende la exactitud de otras aplicaciones como la traducción automática, la búsqueda de respuestas, etcétera.
- Recuperación de información (RI) (Manning, 2007; Baeza & Ribeiro, 1999). Consiste en la búsqueda de documentos de naturaleza no estructurada que satisfaga una necesidad de información considerando grandes colecciones de documentos, por lo general en equipos locales o en Internet. Esta área supera a la búsqueda de bases de datos tradicionales, convirtiéndose en la forma dominante de acceso a la información. Hoy día cientos de millones de personas utilizan, sin saberlo, los sistemas de RI cada día cuando emplean un motor de búsqueda web o hacen búsquedas en sus mensajes de correo electrónico al utilizar, por ejemplo, Google.
- Traducción automática (Gelbukh & Bolshakov, 2003; Bolshakov, *et al*., 2004). Es un sistema que con ayuda de máquinas es responsable de la traducción de un lenguaje a otro. Esta aplicación es muy útil para fines de negocios, comunicación y académicos en razón de que la colaboración internacional crece exponencialmente.
- Búsqueda de respuestas (BR) (Aceves-Pérez, 2007; Ferrández & Ferrández, 2007). Es una tarea compleja que combina las técnicas del PLN, la RI y el aprendizaje automático. El objetivo principal de la BR es localizar la respuesta correcta, a una pregunta escrita en lenguaje natural, dentro de una colección no estructurada de documentos. Los sistemas de BR se parecen a los motores de búsqueda, donde la entrada al sistema sería una pregunta en lenguaje natural y la salida sería la respuesta a dicha pregunta (pero no una lista de documentos enteros como en RI).

- Agrupamiento de documentos (Hernández, García, Carrasco & Martínez 2006). A partir de una colección de documentos de cualquier tema, el objetivo es agruparlos automáticamente considerando los diferentes tópicos que tiene cada uno, de forma que los documentos de un grupo se parezcan mucho entre ellos, pero se diferencien de otros grupos. Este tipo de algoritmos son muy útiles en otras aplicaciones como la GART, donde primero se encuentran los grupos de oraciones o documentos parecidos y luego se resumen. En RI, los documentos recuperados de la consulta son mostrados por grupos, lo que permite ir directamente al grupo de documentos de interés del usuario.

II.1.1 Procesamiento de lenguaje natural en México

Hace ya más de 50 años fueron publicados los primeros avances en el área de PLN a nivel mundial. A partir de ese momento se han realizado una gran cantidad de trabajos en estos temas. Para nuestro país son de especial relevancia las investigaciones realizadas por el Laboratorio de Lenguaje Natural del Centro de Investigación en Computación del Instituto Politécnico Nacional, donde se han creado más de 400 trabajos en los últimos 15 años, principalmente, los realizados y dirigidos por los investigadores Alexander Gelbukh (2014), Grigori Sidorov (2014) e Igor Bolshakov (Bolshakov & Gelbukh, 2000; Bolshakov & Gelbukh, 2004a). En sus obras encontramos las definiciones básicas y los nuevos descubrimientos, producto de la investigación en las diferentes tareas de PLN. Entre otros temas, destacan:

- Recursos léxicos (Gelbukh & Sidorov, 2006)
- Construcción y compilación de diccionarios (Gelbukh, Sidorov, 2002; Gelbukh & Bolshakov, 2003; Gelbukh, Sidorov, Hans & Hernández, 2004a)
- Base de datos de colocaciones (CrossLexica) (Bolshakov, 2000; Bolshakov, 2004b; Bolshakov, Bolshakova, Kotlyarov & Gelbukh, 2008)
- Análisis sintáctico del lenguaje español (Galicia-Haro & Gelbukh, 2007)
- Errores semánticos y malapropismo (Gelbukh & Bolchakov, 2004b; Bolshakov, Galicia-Haro & Gelbukh, 2005)
- Desambiguación del sentido de la palabra (Gelbukh, Sidorov & Han, 2003; Ledo, Sidorov & Gelbukh, 2003)
- Traducción automática (Gelbukh & Bolshakov, 2003)

- Minería de texto (Montes, Gelbukh & López, 2001; Montes-y-Gómez, Gelbukh & López, 2002)

Es justo señalar que, una de las principales conferencias de LC a nivel internacional es organizada por el profesor Alexander Gelbukh (CICLing, 2014). No obstante, otros centros nacionales de investigación han generado nuevos laboratorios de procesamiento de lenguaje natural como en el Instituto Nacional de Astrofísica, Óptica y Electrónica, donde los investigadores Manuel Montes y Gómez (Montes *et al*., 2002), Luis Villaseñor (Denicia, Montes, Villaseñor & Hernández, 2006), José Francisco Martínez Trinidad (Hernández *et al.*, 2006) y Jesús Ariel Carrasco Ochoa (Hernández *et al.*, 2006) han generado o dirigido trabajos relacionados con esta área. Otro de los laboratorios importantes de PLN es el liderado por el investigador Gerardo Sierra Martínez en la Universidad Nacional Autónoma de México. En la Benemérita Universidad Autónoma de Puebla también hay equipos de investigadores que se encuentran trabajando en el área de PLN, sobre todo el liderado por el investigador David Pinto. Lo mismo sucede en la Universidad Autónoma Metropolitana con el investigador Héctor Jiménez, quien también han generado o dirigido trabajos en esta área. De hecho, el interés común y colaboración frecuente de estos investigadores los ha llevado a fundar la Asociación Mexicana para de Procesamiento de Lenguaje Natural (AMPLN) (AMPLN, 2014), a la cual pertenecen también los autores de este libro.

La AMPLN es una organización profesional no lucrativa, con los siguientes objetivos (AMPLN, 2014):

- Promover la interacción e intercambio de las ideas, herramientas y recursos entre especialistas mexicanos en el PLN
- Representar a la comunidad mexicana de especialistas en el PLN
- Difundir los logros y la importancia del PLN en la sociedad nacional

La misión de la AMPLN es fomentar la interacción e intercambio de ideas entre especialistas mexicanos en PLN, así como difundir los logros y la importancia del PLN entre la sociedad nacional.

II.2 Generación de resúmenes de texto

La experimentación a finales de 1950 y 1960 sugería que generar resúmenes de texto por computadora era viable, aunque no sencillo. Tras algunas décadas, los avances en el PLN, junto con la creciente presencia

de texto en línea –en corpus y especialmente en la web–, han renovado el interés en la generación automática de resúmenes de texto. De esta manera, la enorme cantidad de documentos electrónicos disponibles en Internet ha motivado el desarrollo de muy buenos sistemas de recuperación de información. Sin embargo, la información proporcionada por estos sistemas, por ejemplo Google, sólo muestra la parte del texto donde las palabras consultadas aparecen.

De acuerdo con el punto de vista clásico (ver a continuación cómo introducimos nuestro punto de vista), hay tres etapas en la generación automática de resúmenes de texto (Hovy, 2003). La primera etapa se realiza mediante la *identificación de tópicos*, donde casi todos los sistemas emplean varios módulos independientes. Cada módulo hace la asignación de una puntuación a cada unidad de entrada (palabra, oración o pasaje más largo); a continuación, otro módulo combina las puntuaciones de cada unidad para asignar una sola puntuación. Por último, el sistema devuelve las unidades de puntuación más altas, de acuerdo a la longitud del resumen solicitado por el usuario. El rendimiento de los módulos de identificación de tópicos es usualmente medido por las puntuaciones de recuerdo y precisión (véase la siguiente sección).

La segunda ha sido denominada como la etapa de interpretación (Hovy, 2003). Esta etapa distingue los sistemas de generación de resúmenes de tipo extractivo de aquellos de tipo abstractivo. Durante la interpretación los tópicos identificados como importantes son fusionados y representados en nuevos términos, pero también se expresan usando una nueva formulación y utilizando conceptos o palabras no encontradas en el texto original. Ningún sistema puede realizar interpretación sin conocimiento previo sobre el dominio. Por definición, el sistema tiene que interpretar la entrada en términos de algo ajeno al texto. Sin embargo, la adquisición previa del conocimiento de fondo del dominio es tan difícil, que a la fecha los sistemas de GART sólo han tratado una pequeña parte. Por lo tanto, la desventaja en esta etapa es que sigue estando bloqueada por el problema de la adquisición previa del conocimiento de fondo del dominio.

La generación del resumen es la tercera etapa. Cuando el contenido del resumen ha sido creado en una notación interna, entonces requiere de las técnicas de generación de lenguaje natural, a saber: planificación del texto, planificación de oraciones y producción de oraciones.

Desde nuestro punto de vista y del método que proponemos, identificamos cuatro pasos para que un sistema computacional pueda generar un resumen de texto:

Selección de términos. Durante esta etapa uno debe decidir qué unidades contarán como términos, por ejemplo, pueden ser palabras, n-gramas u oraciones.

Pesado o ponderación de términos. Se trata de un proceso de ponderación (o estimación) de los términos individuales con respecto al contenido del documento.

Pesado o ponderación de oraciones. Es el proceso de asignación de una medida numérica de utilidad a la oración. Por ejemplo, una de las maneras de estimar la utilidad de una oración es sumar los pesos de utilidad de los términos individuales de los cuales se compone la oración.

Selección de oraciones. Se seleccionan oraciones u otras unidades como partes finales del resumen. Una de las formas más sencillas para lograrlo es asignar a las oraciones alguna medida numérica que refleje su utilidad dentro del texto original y sólo seleccionar las mejores al elaborar el resumen.

II.2.1 Generación automática de resúmenes extractivos

Sin considerar el pre-procesamiento del texto que normalmente sigue la aplicación de PLN, se presenta a continuación el estado del arte siguiendo las cuatro etapas de un método de GART. El preprocesamiento empleado para esta investigación se presenta en el siguiente capítulo.

II.2.1.1. Selección de términos

La selección de términos más adecuados, de acuerdo con los objetivos planteados en la investigación, debe contener aquellos que sean lo menos dependientes del lenguaje o del dominio. Algunos de estos modelos de selección son la bolsa de palabras (Salton & Buckley, 1988; Salton, 1989), el modelo de *n*-gramas (Villatoro, Villaseñor & Montes, 2006) y las secuencias frecuentes maximales (Ahonen1999, 1999a, 1999b, 2002). Sin embargo, también existen otros modelos interesantes con mayor dependencia como las unidades elementales del discurso (Marcu, 2001; Soricut & Marcu, 2003), factoides (en inglés, *factoids*) (Teufel & Halteren, 2004a, 2004b), bocadillos ricos en información (en inglés, *Information Nuggets*) (Liu, He, Ji & Yang, 2006), unidades de contenido semántico (Nenkova, 2006), cadenas léxicas (Morris & Hirst, 1991) y frases o palabras clave (en inglés, *Keyphrases o Keywords*) (D'Avanzo, Elia, Kuflik & Vietri, 2007). En este caso, consideramos que es adecuado presentar primero los términos que son más dependientes del lenguaje con el objetivo de entender

qué tipo de información tratan de representar. Posteriormente, al presentar los términos menos dependientes del lenguaje se podrá comparar las fortalezas y debilidades de estos modelos.

Las unidades elementales del discurso (UED) fueron utilizadas para la GART en el trabajo de Marcu (2001) y en el de Carlson, Marcu & Okurowski (2003). Las UED son frases o cláusulas que están presentes en las oraciones, las cuales se determinan usando información gramatical, léxica y sintáctica del lenguaje que se esté analizando. Por ejemplo, existen cláusulas preposicionales, sustantivas, verbales, etcétera; las cuales permiten reconocer partes y, en el mejor de los casos, oraciones completas. En los trabajos citados al inicio (Marcu, 2001 y Carlson *et al*., 2003) se utilizaron las cláusulas sustantivas.

Los factoides (en inglés, *factoids*) (Teufel & Halteren, 2004a, 2004b) son unidades semánticas que representan el significado de una oración a través de los hechos que pueden deducirse cuando éstos también se presentan en otros documentos. Por ejemplo, de la oración "*La policía ha arrestado a un hombre holandés blanco*" se pueden derivar los siguientes factoides: "*Un sospechoso fue detenido*", "*La policía hizo el arresto*", "*El sospechoso es blanco*", "*El sospechoso es holandés*" y "*El sospechoso es hombre*". Los factoides son definidos de forma empírica y basados en los datos del conjunto de resúmenes, por lo general algunos de los resúmenes son realizados manualmente tomados del *corpus* (Duc, 2014). La definición de factoide comienza con la comparación de la información contenida en dos resúmenes, es decir, los factoides se agregarán o dividirán, de manera incremental, de acuerdo con los resúmenes que se consideren. Si dos piezas de información ocurren juntas en todos los resúmenes, y en la misma oración, se tratan como un factoide, ya que la diferenciación en más de un factoide no nos ayudaría al distinguir los resúmenes. Los factoides están etiquetados con descripciones en lenguaje natural. Inicialmente, éstos se parecen más a la redacción donde ocurre el factoide en los primeros resúmenes. Aunque el anotador busca identificar y tratar por igual la paráfrasis de la información del factoide cuando éstos ocurren en otros resúmenes. Si, junto con varios enunciados en otros resúmenes, un resumen contiene "*fue asesinado*" y otro "*fue muerto a tiros*", se identifican los factoides: "*Hubo un ataque*", "*La víctima murió*", "*una pistola fue utilizada*". Como se puede ver, es necesario realizar un análisis semántico profundo a una gran cantidad de documentos con el objetivo de obtener factoides relacionados y utilidad.

Otra posible propuesta se presentó por Nenkova (2006) con las unidades de contenido semántico (UCS). La definición de unidad de contenido semántico es algo dinámico, puede ser una sola palabra, pero nunca más grande que una cláusula de oración. La evidencia más impor-

tante de su presencia en un texto es la información expresada en dos o más resúmenes, es decir, es la frecuencia de la unidad de contenido en un texto. Otra evidencia es que estas unidades de contenido frecuente pueden tener una redacción diferente (pero el mismo significado semántico), lo que conlleva dificultades para su extracción independiente del lenguaje.

Las cadenas léxicas, introducidas por Morris y Hirst (1991), explotan la cohesión entre un número arbitrario de palabras relacionadas; su formación se logra mediante el encadenamiento o relación de una de las clases semánticas que tienen las palabras (es decir, tienen un flujo en su sentido) (Barzilay & Elhadad, 1999; Silber & McCoy, 2002). Por ejemplo, como clase semántica se puede utilizar las identidades, sinónimos o los pares hiperónimos/hipónimos, las cuales permitirían agruparlas en la misma cadena léxica. Hay varias dificultades para determinar qué cadena léxica o determinada palabra deben unirse. Por ejemplo, una instancia sustantiva particular puede corresponder a varios sentidos diferentes de la palabra y, por lo tanto, el sistema debe determinar qué sentido debe utilizar. Por ejemplo, si en un caso particular el término "casa" puede interpretarse con el sentido de 1) vivienda o 2) legislatura. Las cadenas léxicas han sido utilizadas en otros trabajos de GART (Brunn, Chali & Pinchak, 2001; Zhou, Sun & Nie, 2005; Li, Sun, Kit & Webster 2007).

Las frases clave (*Keyphrases* en inglés), conocidas también como palabras clave (en inglés *Keywords*), son unidades lingüísticas más grandes que la palabra, pero más cortas que una oración. Hay varios tipos de frases clave que van desde palabras clave de tipo estadístico (sólo secuencias de palabras) hasta las de tipo lingüístico (que se definen en función de una gramática) (D'Avanzo *et al.*, 2007).

Como se puede observar, la mayoría de los términos descritos consideran más de una sola palabra en su definición, tratando de enriquecer semánticamente estos extractos. De igual forma, se puede ver que las diferentes definiciones tratan de limitar o caracterizar cómo es que se pueden presentar estos términos en texto, que, como se vio, requieren de recursos lingüísticos dependientes del lenguaje y del dominio.

En contraste con los términos anteriores, uno de los primeros modelos independientes del lenguaje, y de fácil extracción, es el empleo de las palabras únicas como términos del documento. Este modelo propuesto por Salton, Wong y Yang (1975), Salton y Buckley (1988), y de nuevo por Salton (1989), ahora se conoce como bolsa de palabras (en inglés *Bag of Words*). Sin embargo, al considerar una sola palabra se aumenta la polisemia debido a la pérdida de contexto del propio término. Otro de los problemas que trae consigo este modelo es que existen bastantes términos, incluso para un texto pequeño.

Ante el problema de pérdida de contexto que conlleva la bolsa de palabras, se han utilizado las secuencias de palabras de una longitud predeterminada n, lo que conoce como n-gramas. Aunque el problema de la alta dimensionalidad persiste, los n-gramas se han utilizado ampliamente en diversas investigaciones de GART, debido a su fácil extracción y por disminuir la pérdida de contexto (enriqueciendo sus términos extraídos a mayores tamaños de n), haciéndolos robustos para el problema de la GART (Villatoro, Villaseñor & Montes, 2006).

Con el fin de aumentar la independencia del dominio y del lenguaje con los resúmenes generados, Villatoro *et al*. (2006) eliminan todo tipo de atributos dependientes del lenguaje y del domino, utilizando sólo rasgos basados en palabras. En particular, se utilizan secuencias de palabras (n-gramas) como términos. Aunque en el primer intento al utilizar n-gramas los resultados sobrepasan a los de otros métodos, también se tienen algunas desventajas. Una de ellas es que son siempre secuencias de un tamaño fijo, que se tuvo que definir previamente por el usuario. La mayor parte del problema al utilizar tales técnicas se encuentra en la definición del tamaño de la secuencia a ser extraída, que por lo general depende del análisis del texto.

Otro de los términos interesantes es el propuesto por Ahonen (1999), quien plantea utilizar las secuencias de palabras que son frecuentes, es decir, este modelo necesita un umbral de frecuencia. Dado que las secuencias frecuentes que se pueden extraer son muchas, sólo se queda con aquellas secuencias frecuentes que no son subsecuencias de alguna otra, es decir, que son maximales. Si se habla en términos de n-gramas, equivale a decir que se encuentra el conjunto de todos los n-gramas frecuentes, donde n va desde 1 hasta x (donde x es el tamaño del grama más grande y frecuente), y a partir de ahí el conjunto se reduce sólo a los maximales. Sin embargo, el término secuencia va más allá de un grama, porque no es necesario que sus elementos aparezcan de manera contigua como sucede con el propio grama, pero sí en la misma secuencia u orden. Para restringir la separación que puede haber entre los elementos que forman una secuencia frecuente se utiliza un parámetro conocido como GAP, por sus siglas en inglés. Las secuencias frecuentes maximales (SFMs) con GAP fueron utilizadas en el descubrimiento de tópicos de documentos (Ahonen, 1999a, 1999b, 2002), y en tareas de recuperación de información. En este sentido, se puede ver a las SFMs como frases clave. Aunque parecen tener varias ventajas sobre los n-gramas, una de sus principales desventajas es la complejidad computacional que trae consigo la extracción de SFMs, puesto que crece de acuerdo a como crece el parámetro GAP (García, Martinez & Carrasco, 2004, 2006; García, 2007). Cuando el GAP tiende a ser más grande, la complejidad tiende

a ser exponencial, lo que nos indica que es uno de los problemas más complejos computacionalmente hablando. Si bien, ningún algoritmo (ya desarrollado o que se pueda desarrollar) puede minimizar su complejidad, en el trabajo de García (García *et al.*, 2004, 2006; García, 2007) se desarrollaron algoritmos que permiten manejar de forma más eficiente los recursos computacionales para poder hacer la extracción de SFMs de manera más rápida. Dado que las SFMs parecen ser buenas candidatas para ser utilizadas como términos enriquecidos semánticamente y no habían sido probadas para la tarea de GART, en esta investigación nuestra hipótesis es que podrían mejorar sustancialmente los resúmenes generados de forma automática.

II.2.1.2 Pesado de términos

Los términos identificados en la etapa anterior se ponderan con el fin de seleccionar los más importantes como representantes del texto original.

El uso de la frecuencia como característica de la GART ha demostrado ser útil. La frecuencia del término fue utilizada por primera vez en la GART extractivos en la década de 1950 (Luhn, 1957). Las investigaciones posteriores han utilizado la frecuencia en sus métodos para identificar términos importantes en sus trabajos. Recientemente, el algoritmo SumBasic utiliza la frecuencia del término como parte de un enfoque sensible al contexto para identificar oraciones importantes, al tiempo que disminuye la redundancia de la información (Nenkova & Vanderwende, 2005).

La ponderación propuesta por D' Avanzo *et al*. (2007) se basa en una combinación de la frecuencia del término (en inglés *term frecuency—tf*), por la frecuencia inversa del documento (en inglés, *inverse document frecuency—idf*) y la primera aparición de éste. Es decir, la distancia del término candidato, específicamente las frases clave, es utilizada como término desde el principio del documento en el que aparece. La frecuencia inversa del documento se define como el número de documentos donde aparece el término entre el número de documentos que tiene la colección. Esto hace que los términos sean más importantes cuando aparecen en pocos documentos, lo que permite caracterizar de mejor manera a cada documento. Por ejemplo, las preposiciones ("la", "las", "en", "los", etcétera) o los conectores de conjunciones o disyunciones ("y", "o"), aunque aparecen muy frecuentemente con *tf*, no serían realmente importantes con *idf*, puesto que aparecen en casi todos los documentos.

Nenkova y otros (Nenkova & Passonneau, 2004; Passonneau, Nenkova & Sigleman, 2007; Nenkova, 2006) consideran el pesado de términos a partir de un esquema de pirámide, se trata de un procedimiento

diseñado específicamente para el análisis comparativo del contenido de varios textos. La idea de este esquema consiste en calcular la presencia de cada término en todos los documentos de la colección. Mientras más documentos contengan el término, más importante es y, por lo tanto, se debería incluir en el resumen. Aunque el esquema de Nenkova es aplicado a la GART en múltiples documentos, sería posible aplicarlo a uno solo si en lugar de considerar una colección de documentos se considera la colección de oraciones de un único documento.

Por su parte, Wei *et al.* (2006) derivan la relevancia de un término a partir de una ontología construida con el análisis de conceptos formales. Mientras que Song, Han, y Rim (2004) ponderan una palabra basándose en el número de conexiones léxicas, como las asociaciones semánticas, que la palabra guarda con sus vecinas y se expresan en un diccionario de sinónimos. Junto con esto, las palabras más frecuentes serán ponderadas como más relevantes.

También Mihalcea (2006) presenta una idea similar en forma de red, sustentada en los nodos de un grafo: las palabras que tienen una relación cercana con un mayor número de palabras "importantes" serán importantes por sí mismas. La importancia se estima de manera recursiva en forma similar al algoritmo PageRank (Brin, Page, 2012), el cual es utilizado por Google para ponderar la importancia de las páginas web. La idea es que una oración se pondere como importante si se relaciona con muchas oraciones importantes; la relación se puede entender como el traslape de los contenidos léxicos de las oraciones (Mihalcea, 2006). Los dos métodos presentados por Mihalcea (2006) para la GART reportan algunos de los mejores resultados en la literatura, de ahí que con ellos compararemos nuestro método.

Como se puede observar en los trabajos anteriores, la relevancia de cada término está determinada por la frecuencia de su aparición en el mismo documento-colección de documentos o por la frecuencia de la relación estructural-semántica del contexto del término con sus pares

Cabe señalar que no se han encontrado trabajos modernos donde se utilice las SFMs para el problema de la GART, por lo que no se tiene hasta ahora un esquema de ponderación inicial. Sin embargo, sería posible utilizar la frecuencia y la longitud de cada SFM encontrada en el documento, como dos probables variables que podrían reflejar la importancia de cada término.

II.2.1.3 Pesado o ponderación de oraciones

En la propuesta de Cristea *et al*. (Cristea, 2005) se realiza la ponderación de una oración en función de su proximidad con la idea central del texto, lo cual se determina por análisis de la estructura del discurso.

Sin embargo, las técnicas que tratan de analizar la estructura del texto implican un procesamiento lingüístico demasiado sofisticado y costoso. En contraste, la mayor parte de los métodos descritos en la literatura actual representan al texto y sus frases como una bolsa de características simples, utilizando el procesamiento estadístico sin la intención de "comprender" el texto.

Una muy antigua y simple heurística para el pesado de oraciones no implica ningún término en absoluto, sólo se asigna mayor peso a las primeras oraciones del texto. Textos de algunos géneros –como informes de noticias de periódicos o textos científicos– están diseñados bajo esta lógica. Por ejemplo, casi cualquier artículo científico contiene un resumen al inicio. Esta simple heurística ha mostrado ser muy difícil de superar por otros métodos automáticos, por lo que se ha tomado como línea de base o partida (en inglés, *baseline*) en la colección DUC, 2014 para un nuevo método de la GART. Es decir, la línea de base es la calidad mínima a superar por los nuevos métodos de GART, ya que estarían haciendo un procesamiento "más inteligente", que sólo tomar las primeras oraciones del documento. Vale la pena señalar que en las competiciones de la Conferencia de Entendimiento de Documentos (DUC de las siglas en inglés, *Document Understanding Conference*) (DUC, 2014) sólo cinco sistemas se desempeñaron por encima de esta línea base, lo que no demerita a los otros sistemas, pues esta línea de base es específica del género. Por ejemplo, en contraste negativo, esto no funcionaría en documentos oficiales, mensajes de correo electrónico, páginas web o novelas literarias, puesto que la posición de la oración no estaría disponible para los métodos basados en términos.

Otro de los posibles enfoques para la selección de términos es el de la utilidad relativa, propuesto por Radev, Tam y Erkan (2003). En este enfoque, todas las oraciones de entrada se califican en una escala de 0 a 10, de acuerdo a su aptitud para su inclusión en un resumen. Además, las oraciones que contienen información similar son marcadas explícitamente, por lo que en el cómputo de evaluación se podría sancionar o recompensar la redundancia de oraciones informativamente equivalentes. Este método sólo es aplicable a los sistemas extractivos que seleccionan oraciones de forma directa desde la entrada y no para los sistemas abstractivos.

Por otro lado, está el caso de Verma, Chen y Lu (2007), quienes utilizan el conocimiento de ontologías para la ponderación de oraciones, empleando datos estadísticos de las propias oraciones, así como el análisis

sintáctico de las mismas. La desventaja de esta propuesta es que se hizo para un solo dominio en particular, además requiere de un analizador sintáctico.

II.2.1.4 Selección de oraciones

Los métodos de aprendizaje supervisado consideran la selección de oraciones como una tarea de clasificación. Para esto se entrena un clasificador utilizando una colección de documentos, suministrados con resúmenes existentes. Como características de la oración se consideran los términos seleccionados previamente junto con sus respectivos pesos (Villatoro *et al.*, 2006). Incluso, se han empleado características tanto léxicas como no léxicas (Kupiec, Pedersen & Chen, 1995; Neto, Freitas & Kaestner, 2002; Chuang &Yang, 2004). En el trabajo de Kupiec *et al.* (1995) se propusieron las siguientes características: posición de la oración en el documento, longitud de la oración, presencia de frases clave en la oración y traslape de las palabras de la oración con el título del documento.

Algunos trabajos más recientes (Chuang & Yang, 2004; Neto *et al.*, 2002) extienden estas características incorporando información sobre la ocurrencia de pronombres propios y la presencia de anáforas. Las características denominadas "motivadas heurísticamente" permiten extraer mejores resúmenes. Sin embargo, tienen una desventaja muy grande, ya que pueden ser altamente vinculadas a un dominio específico. Esta condición implica que el cambio de un dominio a otro haga necesario redefinir o incluso eliminar algunas de las características. Por ejemplo, las frases clave, que son particulares para cada dominio, requieren ser modificadas, mientras que el traslape de palabras con el título no tendría sentido en todos los temas, por lo que podría ser eliminado.

Por su parte, Chali y Kolla (2004; Kolla & Chali, 2005) presentaron un trabajo para la GART en múltiples documentos usando cadenas léxicas para la selección oraciones. En este trabajo, el mecanismo de puntuación sólo considera el número de apariciones de las palabras dentro de una oración y dentro de un segmento (un segmento es comparable a documentos únicos dentro de una colección de un solo tema). No se utiliza ninguna información adicional como n-gramas. Los resultados obtenidos indican que la puntuación de una oración basada en cadenas léxicas simples no mejora lo suficiente. Como consecuencia, la ponderación se cambió en el trabajo de Filippova, Mieskes, Nastase, Panzetto & Strube (2007), añadiendo diferentes puntuaciones. Una puntuación agregada fue el número de cadenas que pasan a través de una oración (puntuación de cadena) y la otra puntuación se basa en n-gramas (bigramas y trigramas). Cada ocu-

rrencia de una cadena y un bigrama aumenta la puntuación en uno; un trigrama la aumenta en dos. La puntuación global se calcula para cada frase y luego se ponderan todas las oraciones para la extracción final. Usando el número de cadenas que pasan por una oración se obtienen puntuaciones más altas en oraciones más largas. Esto sucede porque las oraciones más largas también tienen una mayor probabilidad de contener más información, especialmente si varias cadenas pasan a través de ellas.

El enfoque de Seki (2002) se basa en el enfoque de ponderación, calculado de la siguiente manera: en primer lugar se calculan los valores *tf* x *idf* de todos los sustantivos en el documento, exceptuando las palabras vacías (palabras que carecen de significado propio, las cuales se pueden consultar en el anexo A, en inglés *stop-words*). Después, para cada documento se calcula la suma de todos los valores de *tf* x *idf* de los sustantivos. El valor de importancia de una oración se calcula mediante la suma de los valores tf x idf de oraciones que contienen sustantivos. De esta manera se obtiene el valor de importancia resultante para cada oración.

Los enfoques extractivos para GART suelen seguir un modelo de ponderación de oraciones basado en un conjunto de características. Las oraciones con mayor puntuación son extraídas para formar el resumen. Cuando se utiliza la frecuencia como única característica, los elementos del término se cuentan y luego cada oración tendrá una puntuación basada en el cálculo de la frecuencia de cada elemento del término en la oración. Un problema clave en la generación de resúmenes es la disminución de la redundancia. Cada nueva oración en el resumen debería añadir nueva información en lugar de repetir la información que ya está incluida. El uso de los términos más frecuentes probablemente resultará en la misma información repetida varias veces. En el trabajo de SumBasic (Nenkova, & Vanderwende, 2005) se genera primero un modelo de distribución de probabilidad y cada término se utiliza para seleccionar las oraciones. Las probabilidades del término se reducen por lo que los términos con probabilidad más baja tienen una mejor oportunidad de seleccionar oraciones con nuevo contenido de información. Este enfoque se denomina sensitivo al contexto, ya que el sistema de GART considera las oraciones que ya tiene el resumen antes de agregar una nueva oración. Esto también se relaciona con la idea de encontrar relevancia marginal maximal, donde la relevancia marginal se define como la búsqueda de oraciones relevantes que contienen una similitud mínima a las oraciones seleccionadas de manera previa (Carbonell & Goldstein, 1998).

La distribución de frecuencias del algoritmo FreqDist utiliza un enfoque sensible al contexto para ponderar las oraciones basadas en el modelo de distribución de frecuencias, en lugar de un modelo de distribución de probabilidad (Reeve & Han, 2007). El fundamento del enfoque de

distribución de frecuencia sostiene que la distribución de conceptos del texto original debe aparecer en el resumen generado. Es decir, los modelos de distribución de frecuencias del texto original y su correspondiente resumen deberán ser lo más similar posible. Hay dos etapas en el algoritmo: inicialización y generación del resumen. En la etapa de inicialización, los elementos de la unidad (términos o conceptos) del texto de origen se cuentan para formar un modelo de distribución de frecuencias, y se crea una serie de oraciones, llamada serie de oraciones. Un modelo de distribución de frecuencias del resumen se crea a partir de los elementos que se encuentran en el texto original. La cuenta de la frecuencia del modelo de distribución se establece inicialmente en cero para indicar un resumen vacío. En la etapa de generación del resumen nuevas oraciones son evaluadas y luego seleccionadas para incluirse en el resumen. La identificación de la siguiente oración para ser agregada al resumen se lleva a cabo mediante la búsqueda de la oración que más alinee la distribución de frecuencias del resumen generado, hasta el momento, con la distribución de frecuencias del texto original. Para cada oración, en la serie de oraciones se añade la oración al resumen candidato para ver cuánto contribuye al resumen que se quiere obtener. Para determinar la contribución de la oración en la distribución de frecuencias del resumen candidato, se compara la similitud de éste con la distribución de frecuencias del texto de origen. La comparación genera una puntuación de similitud. Esta puntuación se asigna a la oración como la puntuación de la oración. Después de que todas las oraciones (de la serie de oraciones) han sido evaluadas por su contribución a la síntesis de candidatos, se añade la oración con más alta puntuación al resumen y se retira de la serie de oraciones. El proceso de selección de oraciones es iterativo y se repite hasta que se alcanza la longitud deseada para el resumen.

La mayoría de los métodos actuales son puramente heurísticos, no utilizan ningún aprendizaje, sino que directamente establecen el procedimiento utilizado para la selección de términos, el pesado de términos o pesado de oraciones (dado que la selección de oraciones en la mayoría de los casos consiste en la selección de las mejores oraciones ponderadas).

II.2.2 Generación automática de resúmenes abstractivos

Los métodos de GART abstractivos utilizan extracción de información ontológica, información de fusión y de compresión. Estos resúmenes mueven el campo de los métodos puramente de GART extractivos a la generación de resúmenes abstractos con oraciones que no se encontraban en ninguno de los documentos de entrada y que pueden sintetizar información a

través de varias fuentes. Un resumen abstractivo contiene al menos algunas oraciones (o frases) inexistentes en el documento original. Por supuesto, la verdadera abstracción consiste en llevar al proceso un paso más allá. Es decir, implica el reconocimiento de un conjunto de pasajes extraídos, cuya unidad confirma algo nuevo, algo no mencionado de forma explícita en la fuente, que después se reemplaza en el resumen con nuevos conceptos (idealmente más concisos). El requisito de que el nuevo material no esté en el texto significa, explícitamente, que el sistema debe tener acceso a la información externa de algún tipo, como una ontología o una base de conocimientos, donde sea capaz de realizar inferencias combinatorias.

Hay diferentes métodos que se han desarrollado para la GART abstractivos. Por ejemplo, las técnicas de fusión de oraciones (Daume & Marcu, 2004; Barzilay, 2003; Barzilay & McKeown, 2005), la fusión de la información (Barzilay & Elhadad, 1999), compresión de oraciones (Vandeghinste & Pan, 2004; Madnani, Zajre, Dorr, Ayan & Lin, 2007), etc.

II.3 Evaluación de resúmenes automáticos

La ROUGE (Evaluación Suplente de Resúmenes Orientada al Recuerdo) (Lin, 2004a) fue propuesta por Lin y Hovy (Lin & Hovy, 2004; Lin & Och, 2004a, 2004b). Este sistema calcula la calidad de un resumen generado de forma automática mediante la comparación con resúmenes creados por seres humanos. En concreto, se cuenta el número de las diferentes unidades comunes, tales como secuencias de palabras, pares de palabras y n-gramas, entre el resumen a evaluar (el generado por computadora) y los resúmenes ideales creados por seres humanos. La ROUGE incluye varias medidas automáticas de evaluación:

ROUGE-N (coocurrencia de *n*-gramas). Expresa la cobertura o recuerdo de n-gramas entre un resumen candidato y un conjunto de resúmenes de referencia. Se calcula de la siguiente manera:

$$ROUGE-N = \frac{\sum_{O \in \{\text{Resúmenes de referencia}\}} \sum_{grama_n \in O} cuenta_{coincidencia}(grama_n)}{\sum_{O \in \{\text{Resúmenes de referencia}\}} \sum_{grama_n \in O} cuenta(grama_n)},$$

Donde n es la longitud del n-grama y $cuenta_{coincidencia}$ (grama n) el número máximo de n-gramas que co-ocurren en el resumen candidato y en el conjunto de resúmenes de referencia.

ROUGE-L (subsecuencia más larga): Una secuencia S = (s1, s2, ..., sn) es una subsecuencia de otra secuencia X = (x1, x2, ..., xm), si existe una

estricta secuencia en aumento (i1, i2 , ..., ik) de los índices de X tal que para todo j = 1, 2, ..., k, existe xij = sj. Dadas dos secuencias X e Y,la subsecuencia común más larga (SCL) de X e Y es la subsecuencia común con longitud máxima. Cuando SCL se aplica en la evaluación de resúmenes, una oración del resumen es vista como una secuencia de palabras. Intuitivamente, la SCL de dos oraciones es la más similar de dos resúmenes X e Y, donde X es de longitud m e Y de longitud n, suponiendo que X es una oración del resumen e Y es una oración del resumen candidato.

ROUGE-W (subsecuencia ponderada o pesada más larga): Dadas dos secuencias X e Y, SCL se llama ponderada o pesada si la longitud es calculada usando una función de pesado (para más detalles sobre la función ponderada ver Lin & Hovy, 2004).

ROUGE-S (coocurrencia de bigramas no contiguos): Un bigrama no contiguo es cualquier par de palabras, en el orden de la oración, que permite un número arbitrario de espacios. La coocurrencia de bigramas no contiguos mide estadísticamente la cobertura de los bigramas no contiguos, entre el resumen candidato y el conjunto de resúmenes de referencia.

Se mostró en el trabajo de Lin (Lin & Hovy, 2003) que este tipo de medidas se pueden aplicar para la evaluación de la calidad de los resúmenes generados automáticamente, ya que lograron el 95 % de correlación entre juicios humanos.

Para cada una de las medidas de ROUGE (ROUGE-N, ROUGE-L, ROUGE-W, etc.) se calculan las medidas de Recuerdo, Precisión y F-measure como sigue (ver el ejemplo en el Anexo B):

Precisión (*P*): Refleja la cantidad de buenas oraciones extraídas por el sistema:

$$P = \frac{\#(oraciones\ correctas)}{\#(oraciones\ correctas + oraciones\ incorrectas)}$$

Recuerdo: (*R*): Refleja la cantidad de oraciones correctas que olvidó el sistema:

$$R = \frac{\#(oraciones\ correctas)}{\#(oraciones\ correctas + oraciones\ no\ extraídas)}$$

F-measure (F):

$$F = \frac{2PR}{P+R},$$

Donde se toma a oraciones correctas como el número de oraciones extraídas por el sistema y por los humanos, a oraciones incorrectas

como el número de oraciones extraídas por el sistema, pero no por el ser humano, y oraciones no extraídas como el número de oraciones extraídas por el ser humano, pero no por el sistema.

II.4 Evaluación de las herramientas comerciales y métodos del estado del arte para la generación de resúmenes extractivos individuales

En esta sección se evalúa la calidad de siete herramientas comerciales y tres métodos del estado del arte de la GART, utilizando la colección de noticias DUC-2002 y la misma herramienta de comparación de ROUGE descrita en la sección anterior.

II.4.1 Descripción de herramientas comerciales

Las herramientas comerciales se pueden clasificar en instalables y en línea, según el lugar de su ejecución. Las herramientas comerciales en línea de GART son Svhoong Summarizer (Svhoong, 2013), Pertinence Summarizer (Pertinence, 2010), Tool4noobs Summarizer (Tool4noobs, 2013) y Open Text Summarize (Ots, 2013), y dos herramientas comerciales Copernic Summarizer y Microsoft Office Word, en sus versiones 2003 y 2007. A continuación se describen estas herramientas.

La herramienta Shvoong (Shvoong, 2013) fue fundada en 2005 por Avi Shaked y Avner Avrahami. Shvoong es una herramienta que permite generar resúmenes automáticos en 21 idiomas diferentes (checo, neerlandés, danés, inglés, finlandés, francés, alemán, griego, hebreo, húngaro, indonesio, italiano, malayo, noruego, polaco, portugués, rumano, ruso, español, sueco y turco). A diferencia de otras herramientas Shvoong no devuelve el resumen como tal, sino que subraya el texto que considera más importante del documento original.

Pertinence Summarizer (Pertinence, 2010) pertenece a la gama de productos desarrollados con tecnología denominada KENiA© (basada en la extracción de conocimiento y arquitectura de notificación), desarrollada por la empresa francesa Pertinence Mining. Pertinence es una herramienta en línea que permite generar resúmenes en 12 idiomas (alemán, inglés, árabe, chino, coreano, español, francés, italiano, japonés, portugués, ruso y neerlandés) de los documentos de texto en formatos diversos (html, pdf, doc, rtf y txt).

Tools4Noobs (Tool4noobs, 2013) es una herramienta en línea que permite generar resúmenes desde 1 al 100 % del texto original. Para la generación de un resumen Tools4Noobs tiene 3 fases: extracción de las ora-

ciones, identificación de las palabras claves del texto, contando la relevancia de cada palabra e identificación de las oraciones de acuerdo a las palabras claves identificadas.

Open Text Summariser (Ots, 2013) es una aplicación de código abierto para resumir textos que puede descargarse gratuitamente de Internet. También puede encontrarse la interfaz de ésta en línea. OTS genera resúmenes automáticos en diferentes porcentajes y puede generar resúmenes en 37 idiomas.

Copernic Summarizer es un *software* diseñado exclusivamente para la tarea de GART, el cual trabaja con cuatro lenguajes (inglés, alemán, francés y español) (Copernic, 2013).

Microsoft Office Word es una suite ofimática para el procesamiento y edición de texto que incluye la opción de GART.

II.4.2 Descripción breve de los métodos del estado de arte

Se comparó con los métodos heurísticos llamados Baseline (ver la subsección II.2.1.3) y Baseline: aleatorio (ver la Sección V.3.3), los cuales sirven de referencia para medir el avance que presentan los métodos del estado del arte. Baseline es una heurística que consiste en tomar las primeras oraciones del texto para generar el resumen (Villatoro *et al.*, 2006). Baseline: aleatorio es una heurística cuyo funcionamiento consiste en tomar algunas oraciones del texto al azar. Por lo que cualquier método que se comporte como éste no tendría razón de ser. El método que se toma como referencia, por ser uno de los que mejores resultados ha reportado, fue el de TextRank (Mihalcea, 2004a), el cual está basado en la ponderación de grafos.

II.4.3 Configuración de la evaluación

Para poder realizar la comparación entre las herramientas en línea y los métodos del estado del arte de GART se utilizará la colección *Document Understanding Conference* (DUC, 2014). La colección DUC-2002 fue creada por National Institute of Standars and Technology *(NITS)* para el uso de los investigadores en el área de generación de resúmenes. Está compuesta por 567 noticias en inglés, de diversas longitudes y sobre diferentes temas. Cada noticia de DUC-2002 tiene dos resúmenes de 100 palabras creados por dos expertos humanos.

Para evaluar los resúmenes generados automáticamente por las herramientas comerciales se utiliza ROUGE 1.5.5., descrita en la sección anterior. La evaluación consiste en estimar el parecido de los resúmenes

generados automáticamente con los dos resúmenes realizados por los expertos humanos. Los resúmenes generados por las herramientas comerciales instalables y las herramientas en línea fueron generados con un mínimo de 100 palabras, por lo que se analizó cada herramienta para satisfacer la longitud mínima del resumen automático.

II.4.4 Evaluación de las herramientas comerciales en línea e instalables para la GART de un documento

Se compararon cuatro herramientas en línea y dos herramientas instalables. En la figura II.1 se puede observar que las herramientas en línea Shvoong y OTS obtuvieron mejores resultados que las herramientas de Microsoft Office Word. Sin embargo, las otras herramientas en línea Tools4Noobs y Pertinence fueron las que obtuvieron los más bajos resultados. No obstante, el mejor resultado lo obtuvo la herramienta instalable Copernic Summarizer.

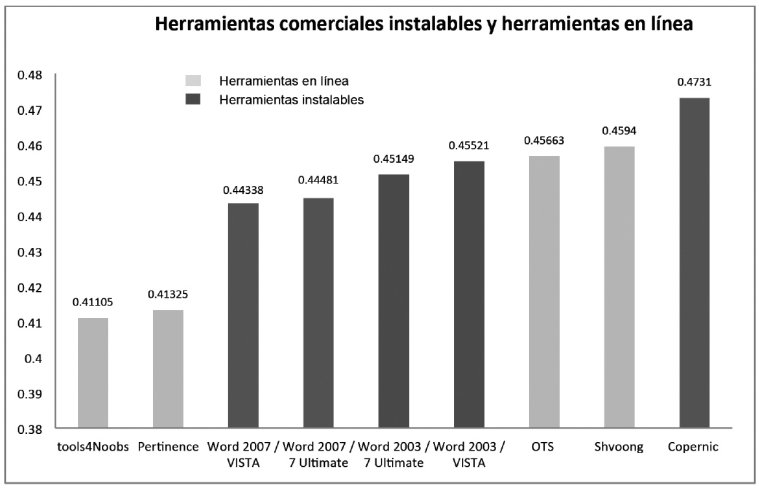


Figura II.1 Comparación de las herramientas comerciales instalables y en línea.

Los resultados obtenidos con Microsoft Office Word instalado en Windows 7 Ultimate no superaron a los resúmenes en las versiones 2003 y 2007, en el sistema operativo Windows Vista. Por lo que el valor que se considerará para la comparación de la herramienta instalable Microsoft Office Word con los métodos del estado del arte y las herramientas en línea, será el obtenido con la versión 2003 y el sistema operativo Windows Vista.

II.4.5 Evaluación de las herramientas comerciales y los métodos del estado del arte

Con el objetivo de conocer el avance que han tenido las herramientas comerciales, en comparación con los métodos del estado del arte, se incluyeron los resultados anteriores junto con siete métodos del estado del arte en la figura II.2.

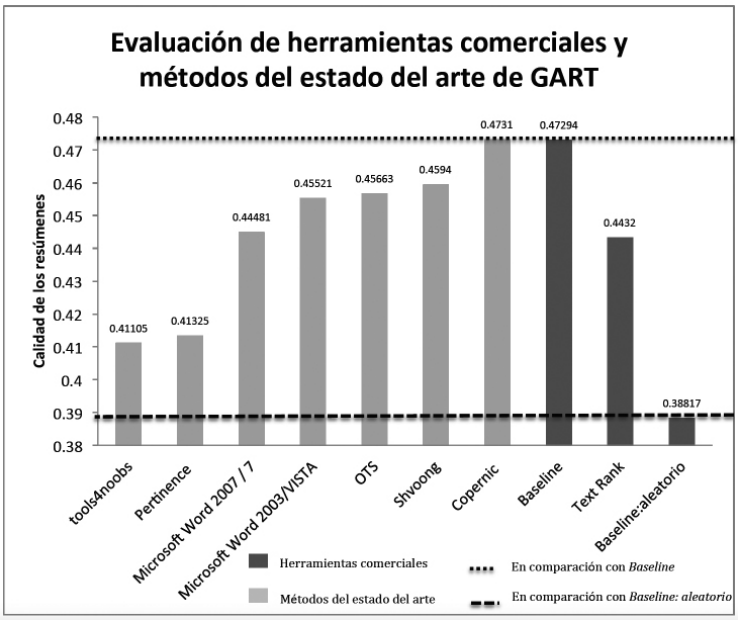


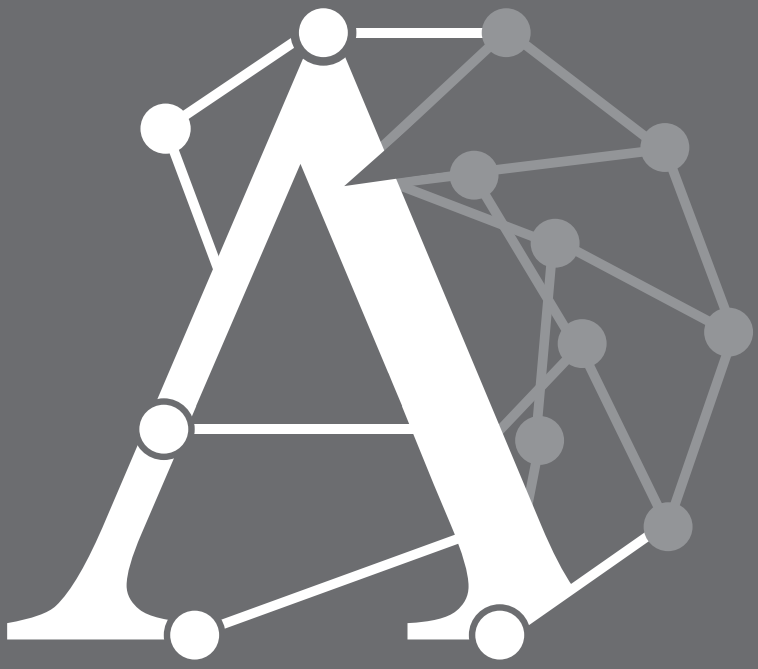
Figura II.2 Evaluación de herramientas comerciales y métodos del estado del arte de GART.

En la figura II.2 se puede observar que los resultados de las herramientas comerciales están por debajo de algunos métodos propuestos en el estado del arte. Es decir, los métodos del estado de arte son de buena calidad.

Una de las heurísticas a superar por las herramientas comerciales instalables y en línea, así como por los métodos propuestos en los estados del arte, es Baseline. Como se puede observar en la figura II.2, sólo Copernic Summarizer (herramienta comercial instalable) supera esta heurística.

Cabe mencionar que aunque las herramientas en línea no superan a la heurística de Baseline, algunas de ellas, como OTS y Shvoong, están por encima de los resultados de la herramienta Microsoft Office Word y de los métodos del estado del arte TextRank y SFM (K-best).

En particular, se encontró que de las cuatro herramientas en línea para GART la mejor fue Shvoong Summarizer. Sin embargo, ninguna de las cuatro herramientas en línea de GART superó a la heurística Baseline. Como resultado de la comparación de todas las herramientas comerciales, tanto herramientas en línea como instalables, se observó que Copernic Summarizer sigue siendo la única herramienta comercial de GART que supera a Baseline.



CAPÍTULO III.
MARCO TEÓRICO

En el capítulo anterior se analizaron por etapas las diferentes combinaciones utilizadas por los métodos del estado del arte para la GART. En este capítulo se van a presentar las definiciones básicas de los modelos que formarán parte de los métodos propuestos en el siguiente capítulo. En la sección III.1 se verán los métodos utilizados para el preprocesamiento del texto que normalmente siguen las aplicaciones del PLN. En la sección III.2 se muestran las definiciones y ejemplos de los términos que se pueden seleccionar con independencia del lenguaje y del dominio. En particular se presenta la definición formal de una secuencia frecuente maximal, dado que los métodos propuestos emplearán este tipo de término (ver subsección III.2.3). En la sección III.3 se aumentan los pesados de términos que comúnmente se vienen empleando en RI y GART. En la sección III.4 se describen, de forma general, los tres tipos de métodos propuestos para la ponderación y selección de oraciones. En la sección III.5 se introducen los algoritmos genéticos, en particular el algoritmo genético básico, así como la representación de población, la función de aptitud y los operadores genéticos.

III.1 Preprocesamiento de texto

La etapa de preprocesamiento depende de qué tan eficiente es la representación de un texto, asunto importante en el área GART, pues incrementa la calidad del resumen obtenido. Los métodos propuestos contemplan la etapa de pre-procesamiento. Por lo general, esta etapa incluirá sólo dos pasos: la eliminación de palabras vacías (ver anexo A) y la lematización de palabras mediante truncamiento (en inglés, *stemming*).

III.1.1 Eliminación de palabras vacías (*stop-words*)

Cuando se realiza el preprocesamiento de un texto se obtiene una representación intermedia del mismo. Una de las etapas de preprocesamiento consiste en la eliminación de palabras vacías del texto. Hay un conjunto de palabras vacías en todos los idiomas, común a todos los dominios, que son fáciles de identificar. Por ejemplo, los artículos, las preposiciones, las conjunciones, etc. Aunque también pueden ser verbos, adjetivos y adverbios. Estas palabras se consideran vacías y normalmente se eliminan.

Las palabras que son muy frecuentes en los documentos dentro de una determinada colección no son buenos discriminadores. De hecho, se considera que una palabra que aparece en al menos el 80 % de los documentos de una determinada colección es inútil al recuperar información. Para el problema de la GART es indispensable eliminar este tipo

de términos puesto que, como se verá en la siguiente sección, presentan una frecuencia muy alta, es decir, los sugiere como términos muy importantes, lo que produciría resúmenes vacíos de información.

Más adelante, realizamos en los documentos un proceso de extracción de palabras vacías, el objetivo es reducir el contenido del texto a las expresiones más específicas (las llamamos descripciones multipalabras), aquellas que contienen sólo las palabras útiles y significativas para la generación de resúmenes automáticos.

Aunque la lista de palabras vacías depende del lenguaje (ejemplo de una lista de palabras vacías en inglés, ver anexo A), la lista no es muy grande, pues habrá alrededor de 200 palabras, además existen varias listas publicadas en diversos lenguajes; por lo tanto, la eliminación de palabras vacías tiene poca dependencia con el dominio y el lenguaje.

III.1.2. Lematización arbitraria de palabras (*stemming*)

Con el objetivo de poder relacionar palabras que significan lo mismo, pero que se escriben ligeramente diferente, por ejemplo, *perro, perros, perra, perrito, perrote*, etcétera, se pueden aplicar los algoritmos de lematización o normalización arbitraria (*stemming*), que principalmente truncan los afijos, prefijos y postfijos de las palabras con el objetivo de tratar de obtener una "raíz" de la palabra, para el caso anterior podría ser *perr*. Se supone que dos palabras que tienen la misma raíz representan el mismo concepto. Básicamente, el proceso de *stemming* se realiza para reducir al mínimo una parte común de una palabra llamada stem, es decir, la parte de la palabra que se queda después de la eliminación de sus afijos, prefijos y sufijos. Los algoritmos de *stemming* tratan de simular a los lematizadores basados en diccionarios, los cuales buscan las raíces lingüísticas de cada una de las formas posibles de una palabra en diccionarios. Sin embargo, para los objetivos de esta investigación, la técnica de *stemming* es deseable dado que requiere de menos recursos lingüísticos. Otra de las ventajas del *stemming* es que puede trabajar con palabras no definidas de forma previa o escritas incorrectamente. Esto simplificará las representaciones de los documentos mediante los modelos mencionados más arriba.

El primer algoritmo de *stemming* fue desarrollado para el idioma inglés y luego fue adaptado para el idioma español. El algoritmo de Porter (1980) es el más utilizado para el idioma inglés. También hay algoritmos para otros idiomas como el francés, el holandés, el griego y el latín. En general, estos algoritmos se basan en un simple conjunto de reglas

que truncan palabras para obtener una raíz en común (Baeza-Yates & Ribeiro, 1999).

III.2. Modelos de selección de términos

Por su prácticamente nula dependencia con el lenguaje y el dominio, los modelos de selección de términos considerados en esta investigación son la bolsa de palabras (Salton *et al.*, 1975), n-gramas (Villatoro-Tello *et al*., 2006) y SFMs (Ahonen-Myka, 1999).

III.2.1 Bolsa de palabras

La representación con la bolsa de palabras (en inglés, *bag of words*) consiste en obtener todas las palabras diferentes que aparecen en el texto. Posteriormente, el documento estará representado por el conjunto de palabras, perdiendo su orden secuencial. El modelo de bolsa de palabras es de fácil extracción, sin embargo, al considerar una sola palabra se aumenta la polisemia debido a la pérdida de contexto del propio término. Otro de los problemas que trae consigo este modelo es que existen muchos términos, incluso para un texto pequeño.

Por ejemplo, para la colección de cinco documentos de la figura III.1 se obtendrían 19 términos, lo cuales serían: *el*, *gobierno*, *de*, *Egipto*, *protege*, *las*, *pirámides*, *son*, *un*, *patrimonio*, *cultural*, *fueron*, *construidas*, *por*, *los*, *faraones*, *tumbas*, *para* y *buen*.

F. el gobierno de Egipto protege las pirámides
G. las pirámides de Egipto son un patrimonio cultural
H. las pirámides fueron construidas por los faraones
I. las pirámides de Egipto fueron tumbas para los faraones de Egipto
J. un buen gobierno protege su patrimonio cultural

Figura III.1 Ejemplo de 5 oraciones de un texto arbitrario.

III.2.2 N-gramas

Ante el problema de pérdida de contexto, que trae la bolsa de palabras, se han utilizado las secuencias contiguas de palabras de un tamaño n predeterminado, lo que se conoce como n-gramas. El modelo n-gramas sigue el mismo principio que el modelo basado en la bolsa de palabras,

la diferencia es que el tamaño del grama, es decir n, es el número de elementos consecutivos que contiene el término. De hecho el modelo de bolsa de palabras se puede representar como 1-gramas. Por ejemplo, si n es igual a 2, el término definido contendrá 2 palabras consecutivas del texto original (también llamado bigramas).

Considerando la colección de documentos de la figura III.1, se obtendrían los siguientes 25 bigramas: *el gobierno, gobierno de, de Egipto, Egipto protege, protege las, las pirámides, pirámides de, Egipto son, son un, un patrimonio, patrimonio cultural, pirámides fueron, fueron construidas, construidas por, por los, los faraones, Egipto fueron, fueron tumbas, tumbas para, para los, faraones de, un buen, buen gobierno, gobierno protege* y *protege su*.

De igual forma que el modelo de bolsa de palabras, los términos extraídos no conservan por completo el orden en que aparecen en el texto. Además, existe otro inconveniente común para ambos modelos: la alta dimensionalidad. Es evidente que, incluso con un pequeño documento, se tiene una cantidad considerable de diferentes características a evaluar, lo que supone un enorme gasto de recursos para manejar tal cantidad de información.

Si bien, el modelo de bigramas o trigramas normalmente mejora la representación semántica de los términos para varias tareas de PLN, no es claro cuál sería el tamaño ideal para cada aplicación, domino o lenguaje.

III.2.3 Secuencias frecuentes maximales (SFMs)

Tratando de resolver los problemas de orden secuencial y dimensionalidad de los modelos, las SFMs se han propuesto para ser utilizadas como un modelo de representación del texto (Ahonen, 1999).

Una secuencia frecuente (SF) es una secuencia donde las palabras aparecen en el mismo orden secuencial y de manera repetida. Una SF se denomina como maximal si no está contenida en otra SF. El modelo de SFMs determina el número de veces que la SF se repetirá en el texto para ser considerada frecuente, este número se denomina umbral de frecuencia.

Es importante tener en cuenta la gran cantidad de secuencias frecuentes que se pueden formar a partir de una colección de documentos pequeña. Por ejemplo, considerando las cinco oraciones del documento de la figura III.1, con un umbral de frecuencia 2, se pueden generar las siguientes 18 SFs contiguas.

"gobierno"	"fueron"
"de"	"los"[1]
"Egipto"	"faraones"
"protege"	"de Egipto"
"las"[2]	"las pirámides"
"pirámides"	"pirámides de"
"un"	"patrimonio cultural"
"patrimonio"	"los faraones"
"cultural"	"las pirámides de Egipto"

A partir de la lista anterior se puede observar por qué es importante quedarse únicamente con las SFMs.

Definición de manera formal:

Una secuencia *S*, denotada por $<s_1,s_2, \ldots,s_k>$, es una lista ordenada de *k* elementos. Una secuencia de longitud *k* es denominada k~secuencia.

Sean $P=<p_1, p_2, \ldots, p_t>$ y $S= <s_1,s_2, \ldots,s_m>$ secuencias, P es una subsecuencia con GAP=0 de *S*, denotado como $P \subseteq_0 S$ si existe un entero $i \geq 1$ tal que

$$p_1= s_i, p_2 = s_{i+1}, p_3 = s_{i+2}, \ldots, p_t= s_{i+(t-1)}$$

Una oración *W* de un documento D se puede considerar como una secuencia de palabras, denotado también como

$$<w_1, w_2, \ldots, w_t>$$

La frecuencia de una secuencia *S* en una colección de oraciones $\{W_1, W_2, \ldots, W_j,\}$, consideradas como secuencias, se denota por S_f o $<s_1,s_2, \ldots,s_t>_f$ que es el número de oraciones en las cuales *S* aparece por lo menos una vez, esto es, $S_f = |\{W_i | S \subseteq_0 W_i\}|$.

Dado un umbral definido por el usuario (β), una secuencia *S* es frecuente si $S_f \geq \beta$. Una secuencia frecuente *S* es maximal si *S* no es subsecuencia de alguna otra secuencia frecuente.

Una colección de oraciones $\{W_1, W_2, \ldots, W_j,\}$ forma un documento *D* cuando $\{W_1, W_2, \ldots, W_j,\} \in D$

Dado un documento se pueden extraer todas las secuencias frecuentes maximales, considerando el GAP y el umbral β.

Por ejemplo, el documento presentado en la figura III.1 está compuesto por cinco oraciones a partir de las cuales con β =2 y GAP=0, se encontrarían como SFMs: *un, gobierno, fueron, protege, patrimonio cultural, los faraones* y *las pirámides de Egipto*.

Utilizando el mismo documento presentado en la figura III.1, con β =2 y GAP=2, se encontrarían como SFMs: *un, gobierno protege, las Egip-*

[1] "*Los*", plural masculino.

[2] "*Las*", plural femenino.

to, patrimonio cultural, las pirámides Egipto, las de Egipto, las pirámides de Egipto y *las pirámides fueron los faraones*. Se puede ver más ejemplos de SFMs a partir de la colección DUC-2002 en el Anexo C.

III.3 Pesado o ponderación de términos

El pesado o ponderación de términos consiste en asignar un valor para que cada término refleje su importancia con respecto a los otros. Como se vio en el capítulo anterior, las ponderaciones de términos independientes del lenguaje están basadas en la frecuencia que el término tiene en el documento. A continuación se describen las ponderaciones de términos más comunes.

Pesado booleano. Es la forma más fácil de pesar un término. Se asigna el valor de 1 si aparece en la oración y valor 0 en otro caso.

$$p_i(t_j)=\begin{cases}0, & si \quad aparece \\ 1, & en \quad otro\ caso\end{cases} \tag{3.1}$$

Donde t_j es la frecuencia de término *j* que tiene en la oración p_i. En la tabla III.1 se muestra el pasado booleano correspondiente a las oraciones de la figura III.1, utilizando como términos a las SFMs con β =2 y GAP=0.

Tabla III.1 Pesado booleano basado en las SFMs con β =2 y GAP=0 para las oraciones de la figura III.1

SFMs \ Documento	A	B	C	D	E
Un	0	1	0	0	1
Fueron	0	0	1	1	0
Gobierno	1	0	0	0	1
Protege	1	0	0	0	1
patrimonio cultural	0	1	0	0	1
los faraones	0	0	1	1	0
las pirámides de Egipto	0	1	0	1	0

Frecuencia de término, abreviado *tf* por sus siglas en inglés (*term frequency*). Esta ponderación, propuesta por Luhn (1957), considera que

un término repetido con frecuencia en una oración puede reflejar mejor el contenido de ésta, en comparación de aquel que se repite menos veces. Por lo tanto, el pesado *tf* asigna un peso mayor a los términos con mayor frecuencia y consiste en evaluar el número de veces que la palabra aparece en la oración.

$$p_i(t_j) = f_{ij} \tag{3.2}$$

Donde, f_{ij} es la frecuencia de término *j* en la oración *i*. En la tabla III.2 se muestra el pasado por frecuencia correspondiente a las oraciones de la figura III.1, utilizando como términos a las SFMs con β =2 y GAP=0.

Tabla III.2 Pesado por frecuencia, basado en las SFMs con β =2 y GAP=0 para las oraciones de la figura III.1

SFMs \ Documento	A	B	C	D	E
Un	0	1	0	0	1
Fueron	0	0	1	1	0
Gobierno	1	0	0	0	1
protege	1	0	0	0	1
patrimonio cultural	0	1	0	0	1
los faraones	0	0	1	1	0
las pirámides de Egipto	0	1	0	1	0

Frecuencia de documento inversa, abreviado *idf* por sus siglas en inglés (*inverse document frequency*). Propuesta por Salton y Buckley (1988), considera que un término muy frecuente en varios documentos es menos útil que uno cuya frecuencia es menor pero se presenta en pocos documentos. Lo que evalúa esta medida es la distribución de los términos en el documento y se define como:

$$p_i(t_j) = \log(\frac{N}{n_j}) \tag{3.3}$$

Donde f_{ij} es la frecuencia del término *j* en el documento *i*; *N* es el número de documentos en la colección; n*j* es el número de documentos en los que aparece el término *j*.

Cabe señalar que esta medida se puede aplicar al problema de la GART en un solo documento si se considera una colección de oraciones y no una colección de documentos.

En la tabla III.3 se muestra el pesado por frecuencia inversa del documento correspondiente a las oraciones (tomadas como documentos) de la figura III.1, utilizando como términos a las SFMs con β =2 y GAP=0.

Tabla III.3 Pesado por frecuencia inversa del documento, basado en las SFMs con β =2 y GAP=0 para las oraciones de la figura III.1

SFMs \ Documento	A	B	C	D	E
Un	0	0.398	0	0	0.398
Fueron	0	0	0.398	0.398	0
Gobierno	0.398	0	0	0	0.398
protege	0.398	0	0	0	0.398
patrimonio cultural	0	0.398	0	0	0.398
los faraones	0	0	0.398	0.398	0
las pirámides de Egipto	0	0.398	0	0.398	0

Pesado tf-idf. Es común que la frecuencia de los términos (tf) y la frecuencia inversa del documento (idf) se utilicen juntas, el fin es determinar la relevancia de cada término, considerando tanto la importancia que tiene el término en la colección de documentos como su importancia en ese documento (Salton & Buckley, 1988). Esta combinación se conoce como ponderación *tf* x *idf* y consiste en multiplicar la frecuencia del término, con respecto al documento, por la frecuencia inversa del documento en los que aparece.

$$p_i(t_j) = f_{ij} \times \log\left(\frac{N}{n_j}\right) \tag{3.4}$$

Es importante tomar en cuenta que se está trabajando con un único documento, por lo que se tomará a *N* como el número de oraciones y n_j, como el número de oraciones en las que aparece el término. Siguiendo el ejemplo de la colección de documentos de la figura III.1, con el pesado *tf-idf* se obtendrían los mismos pesos de la tabla III.3, esto sucede porque la colección de documentos es muy pequeña por lo que es difícil que haya una SFM en más de una oración.

Ponderación de acuerdo a la longitud. Como se vio en la sección anterior, los términos con descripciones multipalabras enriquecen su

contenido semántico, por ello se busca integrar en los términos a varias palabras que aporten significado. Las SFMs, a diferencia de los términos de bolsa de palabras y n-gramas, varían la cantidad de palabras que traen consigo. De ahí que en esta investigación se utilice la longitud de la SFM como medida de relevancia del término. Es decir:

$$p_i(t_j) = |t_j| \tag{3.5}$$

En la tabla III.4 se muestra el pesado por longitud de la secuencia correspondiente a la colección de documentos de la figura III.1, utilizando como términos a las SFMs con β =2 y GAP=0.

Tabla III.4 Pesado por longitud, basado en las SFMs con β =2 y GAP=0 para la colección de documentos de ejemplo de la figura III.1

SFMs \ Documento	A	B	C	D	E
Un	0	1	0	0	1
Fueron	0	0	1	1	0
Gobierno	1	0	0	0	1
Protege	1	0	0	0	1
patrimonio cultural	0	2	0	0	2
los faraones	0	0	2	2	0
las pirámides de Egipto	0	4	0	4	0

III.4 Pesado y selección de oraciones

El pesado de oraciones tiene como última etapa la selección de oraciones, por lo que estas dos etapas se han trabajado juntas por diversos algoritmos. Por ejemplo, algoritmos de agrupamiento (ver sección IV.5), genéticos (García & Ledeneva, 2013) y de ponderación de grafos (Mihalcea, 2004). Como se comentó en la sección del estado del arte, el algoritmo TextRank (Mihalcea, 2006), adaptación del algoritmo de PageRank que pondera páginas web para utilizarlas en Google, ha mostrado ser un buen método para pesar y seleccionar las oraciones que formarán el resumen.

III.4.1 Algoritmo *TextRank*

En esta sección se explica cómo funciona el algoritmo TextRank. Para realizar la aplicación de algoritmos de ponderación, basados en grafos para textos en lenguaje natural, se construye un grafo que representa el texto e interconecta palabras u otras entidades de texto con las relaciones significativas. Los grafos construidos de esta manera se centran en el texto fuente, pero se pueden ampliar con los grafos externos, como las redes semánticas o asociativas u otras estructuras similares derivadas automáticamente de grandes corpus.

Los nodos o vértices del grafo se definen en función de la aplicación, se pueden añadir diferentes segmentos del texto los cuales pueden tener diferentes tamaños y características. Por ejemplo, palabras, sentidos de palabras, oraciones enteras, documentos completos u otros. Cabe notar que los nodos del grafo no tienen que pertenecer a la misma categoría.

Las aristas o arcos entre los nodos del grafo se determinan de acuerdo al tipo de relación que hay entre los nodos. Por ejemplo, las relaciones léxicas o semánticas, las medidas de cohesión del texto, la superposición contextual, la pertenencia a una palabra en una oración, entre otras.

Algoritmo: Sin importar el tipo y las características de los elementos añadidos al grafo, la aplicación de los algoritmos de ponderación de textos se componen de los siguientes pasos:

1. Identificar los segmentos de texto que mejor definen la tarea en cuestión y agregarlos como vértices del grafo.
2. Identificar las relaciones que conectan dichas unidades de texto y utilizar estas relaciones para crear aristas entre vértices en el grafo. Las aristas pueden ser dirigidas o no dirigidas, ponderadas o no ponderadas.
3. Aplicar un algoritmo de ponderación basado en grafos para encontrar la relevancia en los nodos del grafo. Iterar el algoritmo de ponderación basado en grafos hasta que converja. Ordenar aristas basadas en su puntuación final. Utilizar los valores asociados a cada arista para las decisiones de ponderación/selección.

El algoritmo general TextRank se definió para la tarea de la GART mediante un grafo con aristas no dirigidas ponderadas, de la siguiente manera:

1. Las oraciones del texto original van a formar los vértices del grafo.
2. Los pesos de las aristas que conectan los vértices se calcularán utilizando la similitud del coseno entre las dos oraciones que representan dichos vértices. En este sentido, las oraciones son representadas como bolsas de palabras con el modelo booleano. Es decir, los vértices X_i y Y_j, correspondiente a los documentos D_i y D_j, estarán representados por los conjuntos $(x_{1i}, x_{2i}, ..., x_{mi})$ y $(x_{1j}, x_{2j}, ..., x_{mj})$, respectivamente. La similitud del coseno entre ambos vectores se calcula como:

$$\cos(D_i, D_j) = \frac{\sum_{l=1}^{m} x_{li} x_{lj}}{\sqrt{\sum_{l=1}^{m} x_{li}^2 \sum_{l=1}^{m} x_{lj}^2}} \tag{3.6}$$

3. Calcular la relevancia de los vértices se logra mediante un proceso iterativo recursivo hasta que convergen todos los pesos (es decir, ya no hay cambios en los pesos o es menor a un umbral predefinido). En la primera iteración, para calcular la primera relevancia de cada nodo se puede emplear la suma de sus aristas previamente calculadas. Para las iteraciones posteriores ya no se utilizan los pesos de las aristas sino el peso de los vértices, lo cual se hace hasta que converjan. La puntuación de cada arista se vuelve a calcular en cada iteración sobre la base de los nuevos pesos que las aristas vecinas han acumulado. El algoritmo termina cuando se alcanza el punto de convergencia para todas las aristas, lo que significa que la tasa de error para cada arista cae por debajo de un umbral predefinido. Para ello se utiliza el algoritmo PageRank (Brin & Page, 2012), pero adaptado para grafos no dirigidos con la siguiente fórmula.

Dado un grafo dirigido G compuesto de vértices *(V)* y aristas *(A)*, *G= (V, A)*, donde $In(V_i)$ es el conjunto de aristas que apuntan al vértice V_i (predecesores) y $Out(V_i)$ es el conjunto de aristas que apunta al vértice V_i (sucesores), es decir $A=\{In(V_i), Out(V_i)\}$. El algoritmo PageRank asociado con la arista V_i se define mediante la función recursiva $S(V_i)$ que integra las puntuaciones de sus predecesores:

$$S(V_i)=(1-d)+d* \sum_{V_j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|} \tag{3.7}$$

Donde *d* es el parámetro que se establece entre 0 y 1.

Por ejemplo, para calcular la similitud del coseno entre los vértices que formarían las oraciones *C* y *D,* de la colección de documento, figura III.1, se puede considerar la representación de la tabla III.5, donde $D_i = C$ y $D_j = D$.

Tabla III.5 Representación de bolsa de palabras, ponderadas mediante el esquema booleano, documentos C y D de las oraciones de la figura III.1, empleados para el cálculo de la similitud del coseno entre C y D

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Doc	las	pirámides	fueron	construidas	por	los	faraones	de	Egipto	tumbas	para
D_i	1	1	1	1	1	1	1	0	0	0	0
D_j	1	1	1	0	0	1	1	1	1	1	1

Donde, $\sum_{l=1}^{m} x_{li} x_{lj} = 5$, $\sum_{l=1}^{m} x_{li}^2 = 7$, $\sum_{l=1}^{m} x_{lj}^2 = 9$, por lo que el *cos (C,D)* = 0.63

Realizando todos los cálculos de las aristas que hay entre las oraciones de la figura III.1, se generaría para su primera iteración el grafo mostrado en la figura III.2. En el grafo de la figura III.2 se calcularon los primeros pesos correspondientes a los vértices, donde se puede apreciar la ponderación inicial que tienen las oraciones dentro del documento. En orden de importancia las oraciones serían B, A, D, C y E. Para formar el resumen se van agregando las oraciones más ponderadas hasta cubrir el número de palabras o porcentaje deseado por el usuario.

Una vez aplicado el algoritmo de PageRank al grafo de la figura III.2, el grafo quedaría como el de figura III.3. Esto nos dice que las oraciones ordenadas de acuerdo a su relevancia serían A,B,D,C,E; donde A resultó ser más relevante que B después de aplicar PageRank.

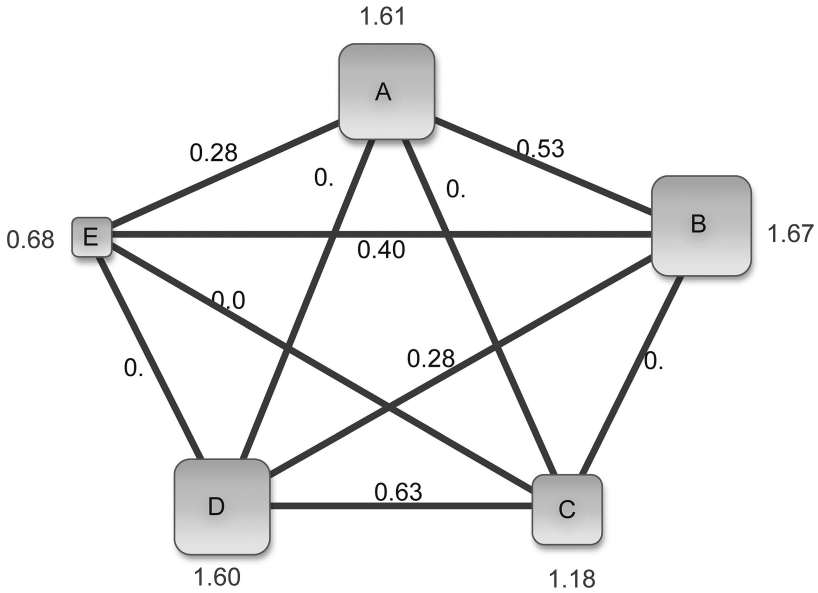


Figura III.2 Representación del grafo utilizado por TextRank (Mihalcea, 2006) para calcular la ponderación de las oraciones de la figura III.1. El tamaño del nodo representa la importancia inicial de la oración dentro del documento.

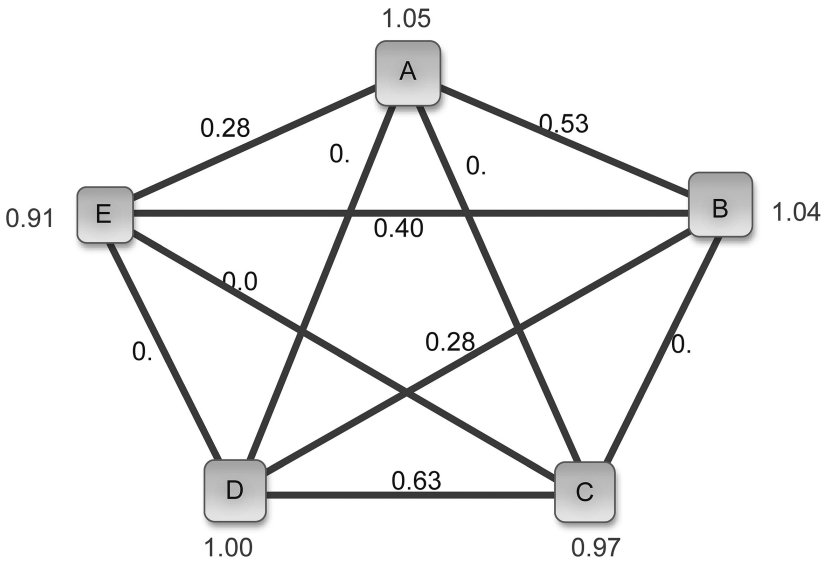


Figura III.3 Representación del grafo resultante por TextRank (Mihalcea, 2006) a las oraciones de la figura III.1. El tamaño del nodo representa la importancia final de la oración dentro del documento.

Con el fin de tener una aproximación tanto cualitativa como cuantitativa a la calidad del resumen generado por TextRank, se solicitó a 10 personas que ordenaran las oraciones del ejemplo de la figura III.1, de mayor a menor importancia según su criterio. Además, se solicitó que escogieran dos oraciones a manera de resumen, sin importar si coincidían con las dos oraciones más importantes antes ordenadas. Para medir cuál sería el orden de las oraciones que en promedio contestaron, se le asignó a la oración más importante la relevancia de 5, a la siguiente oración de 4 y así sucesivamente. Por último, las relevancias de cada oración se promediaron y ordenaron de forma descendente. El resultado fue *D* (3.9) *C* (3.9), *B* (3.3), *E* (2), *A* (1.9), donde el valor entre paréntesis corresponde al promedio obtenido. Como se puede ver, *D* y *C* resultaron tener el mismo orden de importancia. Considerando el resultado anterior como uno de los objetivos que persigue TextRank, se puede ver que las oraciones más importantes (*D* y *C*) fueron ordenadas después de *A* y *B*, sin embargo, *A* fue la peor oración según el criterio humano.

Para tener una medición cuantitativa se podría utilizar la distancia de Manhantan que consiste en sumar el número de posiciones en que difiere cada oración resultante de TextRank (*A, B, D, C, E*) con las posiciones promedio de los encuestados. Como *D* y *C* tienen el mismo peso se tomará entonces la combinación de menor medición. En este caso ambas combinaciones, *C, D, B, E, A* y *D, C, B, E, A* dieron 10. Por ejemplo, considerando *C,D,B,E,A* y *A,B,D,C,E* con la oración *A* existen cuatro posiciones de diferencia entre *C,D,B,E,A* y *A,B,D,C,E*, con la oración *B* una, con la oración *C* tres, con *D* una y con *E* una, la suma da 10.

Como las dos oraciones más importantes que seleccionaron las personas no necesariamente son las oraciones que formarían el resumen (la segunda oración puede ser muy parecida a la primera y no aportar información, por lo que se tendría que escoger otra oración), se preguntó en la encuesta cuáles serían esas dos oraciones. En la encuesta, las oraciones seleccionadas que más votos recibieron de forma descendente fueron *D*(7), *B*(6), *C*(3), *E*(2) y *A*(2). Por lo que, las oraciones *D* y *B* serían las oraciones del resumen, puesto que tienen un puntaje del doble sobre *C*, *E* y *A*. Como puede observarse, los humanos no coinciden del todo en que las oraciones más importantes sean las que deben formar parte del resumen. Por ejemplo, coinciden en que la oración más importante es *D* y que debe formar parte del resumen, pero según los encuestados la oración *B* está en tercer lugar de importancia, no obstante, debería formar parte del resumen. También puede verse que los encuestados coinciden en que la oración *A* es la peor.

En este caso, TextRank para la GART seleccionaría las dos oraciones más ponderadas que serían: *A* y *B*. En este caso, TextRank y los encues-

tados coinciden sólo en la oración *B*. Cabe señalar que la oración mejor ponderada por TextRank fue *A*, sin embargo, fue la peor según los encuestados.

III.5 Optimización de procesos mediante algoritmos genéticos

Un algoritmo genético (AG) utiliza los principios de la evolución, la selección natural y la genética de los sistemas biológicos naturales, representados en un algoritmo de computadora, para simular la generación de soluciones de manera evolutiva en problemas de optimización (Goldberg, 1989). En esencia, el AG es una técnica de optimización que realiza la búsqueda paralela, estocástica, pero dirigida a evolucionar una población haciéndola más apta.

Los AGs codifican una posible solución a un problema específico en una simple estructura de datos, como el cromosoma, y aplican los operadores de recombinación a estas estructuras a fin de preservar la información crítica. Los AGs son a menudo vistos como optimizadores de funciones, por lo cual existen una amplia diversidad de problemas a los que se han aplicado. Las aplicaciones más comunes de AGs son la solución de problemas de optimización, donde han mostrado resultados eficientes y fiables.

La historia de los algoritmos genéticos se remonta a principios de la década de 1970, cuando John Holland (Holland, 1975) introdujo este concepto. Su objetivo era hacer que las computadoras simulen lo que hace la naturaleza. Holland estaba ocupado con los algoritmos que manipulan cadenas de dígitos binarios. Cada "cromosoma" artificial consiste en un número de "genes" y cada gen está representado por 0 o 1:

1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1

La naturaleza tiene la capacidad de adaptarse y aprender sin que se le diga qué hacer. En otras palabras, la naturaleza encuentra buenos cromosomas ciegamente, los AGs hacen lo mismo. Dos mecanismos vinculan un AG para el problema que se está resolviendo: codificación y evaluación. El AG necesita de una medida de aptitud de los cromosomas individuales para llevar a cabo la selección de los mejores individuos y su reproducción (Darwin, 1956). Según Darwin, la selección natural es el proceso mediante el cual los individuos, generalmente los más aptos a su ambiente, logran reproducirse y con ello heredar, a tra-

vés de sus genes, características fuertes. Es decir, al cruzarse tienden a producir mejores individuos. Sin embargo, en el proceso de selección es posible que individuos no tan aptos también logren reproducirse, lo que posibilita, aunque en un porcentaje bajo, obtener mejores individuos cuando uno o los dos individuos no son tan aptos. Otro de los procesos relevantes dentro de la evolución, según Darwin, quizá el más importante, es el enriquecimiento de material genético mediante el mecanismo de mutación. La mutación permite que los descendientes de cierta generación exhiban características inexistentes en los padres, lo cual, aunque poco frecuente, es importante para buscar nuevas combinaciones de cromosomas que posibiliten mejores individuos, esto permite que las poblaciones evolucionen.

III.5.1 Algoritmo genético básico

El algoritmo genético básico consta de los siguientes pasos (Negnevitsky, 2005):

Paso 1. Representar las variables del dominio del problema como un individuo con cromosoma de longitud fija; se elige el tamaño de una población de individuos, la probabilidad de cruce y la probabilidad de mutación

Paso 2. Se define una función de aptitud para medir el rendimiento o aptitud de un cromosoma individual en el dominio del problema. La función de aptitud establece la base para la selección de los individuos que se aparearon durante la reproducción

Paso 3. Se genera una población inicial de individuos de tamaño N: $x_1, x_2, ..., x_N$

Paso 4: Se calcula la aptitud de cada individuo $f(x_1), f(x_2), ..., f(x_N)$

Paso 5. Se selecciona como padres a un par de individuos de la población actual. La selección se realiza de forma probabilística y en relación con su aptitud

Paso 6. Se cruza a los padres para generar un nuevo individuo mediante la aplicación de un operador genético

Paso 7. Algunos de los individuos serán mutados de acuerdo a una baja probabilidad. Se colocan los individuos de descendencia creados en la nueva población

Paso 8. Se repite el paso 5 hasta que el tamaño de la nueva población de individuos sea igual al tamaño de la población inicial

Paso 9. Se eliminan a los padres de la población actual dejando sólo a los descendientes

Paso 10. Ir al paso 4 y repetir el proceso hasta que el criterio de terminación esté satisfecho. El criterio de terminación puede ser que se haya alcanzado un individuo con una determinada adaptación, que se haya alcanzado un número máximo de iteraciones, que la población se haya estabilizado (es decir, que la mayoría de todos sus individuos comparten los mismo genes, por mencionar algunos)

Un AG representa un proceso iterativo. Cada iteración es una generación. Un número típico de generaciones para un AG simple puede variar de 50 a más de 500. Debido a que los AGs utilizan un método de búsqueda estocástica, la aptitud de una población puede mantenerse para un número de generaciones antes de que aparezca un individuo superior.

III.5.2 Representación de población, función de aptitud y operadores genéticos

En esta sección se describen algunos elementos básicos que forman parte del algoritmo genético anteriormente descrito.

III.5.2.1 Representación

Antes de aplicar un AG primero debemos codificar los parámetros del problema a optimizar. Los AGs no tratan directamente con los parámetros, trabajan con los códigos que representan los parámetros. Por lo tanto, la representación del problema es la primera cuestión importante en el diseño de algoritmos genéticos, es decir, cómo representar los parámetros del problema.

Los diferentes esquemas de representación pueden causar diferente desempeño en los AGs (Chambers, 1999; Haupt & Haupt, 2004; Melanie, 1999). En este orden, existen dos métodos de representación utilizados comúnmente: punto flotante y cadena de bits. El método preferido es la cadena binaria, porque la mayoría de los operadores genéticos son adaptados para este tipo de representación, además esta representación tiene un mayor impacto en el rendimiento de los algoritmos genéticos. En la representación binaria de AGs cada parámetro para optimizar se codifica mediante una cadena binaria de longitud fija, por lo que necesitamos encontrar una función de codificación que asigne un valor de parámetro real de un número entero en el intervalo $[0, 2^{l-1}]$, *donde l* es la longitud de la cadena binaria. Para construir una función de este tipo, por lo general, primero se decide el rango de cada

valor del parámetro, basado en el conocimiento previo del problema. Con base en el rango y la precisión deseada del valor óptimo para cada parámetro, se puede calcular la longitud de la cadena binaria requerida. El papel de la función de codificación y su inversa (función de decodificación) es codificar y decodificar un espacio de soluciones posibles para un parámetro, de manera que podemos pasar de valores de los parámetros reales a una cadena binaria útil a los AGs.

III.5.2.2 Población

Los algoritmos genéticos funcionan con una población de posibles soluciones, por lo que, al principio de un AG se requiere una población inicial de individuos. El tamaño de la población inicial puede ser fijo, aunque dependiendo del algoritmo esto puede ser adaptado. Hay tres maneras de formar la población inicial: aleatoriamente, determinista y por ayuda de otros métodos. Los primeros métodos generan soluciones al azar. El segundo inicializa la población con cromosomas específicos, por ejemplo, sólo los cromosomas de 0's, 1's, y así sucesivamente (O'Reilly, Yu, Riolo & Worzel, 2006). También el conocimiento del problema se puede utilizar y obtener soluciones que satisfagan ciertos requisitos. Finalmente, la población inicial también puede ser inicializa con individuos encontrados por otras técnicas de optimización.

III.5.2.3 Función de evaluación de la aptitud

La aptitud de un individuo en los algoritmos genéticos es el valor devuelto por la función de evaluación de la aptitud. Esta función de evaluación mide la calidad de los cromosomas para resolver un problema. Obviamente, la aptitud de los cromosomas menos aptos al resolver un problema se castiga más que la aptitud de los cromosomas más aptos.

La función de evaluación de aptitud actúa como interfaz entre el algoritmo genético y el problema de optimización. En primer lugar, el cromosoma debe ser decodificado y luego evaluado por la función de aptitud para devolverle un valor que indique la aptitud de los cromosomas al resolver el problema. La función de evaluación de la aptitud desempeña un papel importante en el AG, ya que proporciona información acerca de la eficiencia de una solución al resolver el problema. Esta información guía la búsqueda de un algoritmo genético, y con mayor precisión, la evaluación de los resultados de la función de aptitud para

determinar la probabilidad de que una posible solución cree nuevas respuestas en la próxima generación.

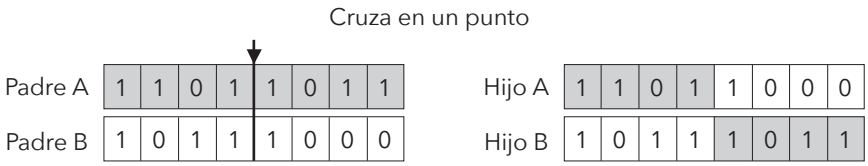
III.5.2.4 Operador de cruza

Cruza es un operador genético que combina dos cromosomas (padres) para producir uno o dos cromosomas (descendiente). La idea detrás del operador de cruza es que el nuevo cromosoma pueda ser mejor que ambos padres si toma las mejores características de cada uno. En primer lugar, el operador de cruza elige al azar un punto en dos cromosomas de los padres y luego intercambia las partes de los cromosomas después de ese punto, con una probabilidad de cruza definida por el usuario. Como resultado, se crean dos nuevas crías. Si un par de cromosomas no se cruza, entonces la clonación de cromosomas se lleva a cabo y, por lo tanto, la descendencia es una copia exacta de cada padre (Negnevitsky, 2005).

Las formas más comunes de cruce son de un punto, de dos puntos, n-punto y la cruza uniforme, todas ellas se muestran en la figura III.4.

III.5.2.5 Operador de mutación

El operador de mutación representa un cambio en el gen (figura III.5). Su función es proveer y garantizar que el algoritmo de búsqueda no esté atrapado en un óptimo local. El operador de mutación utiliza la probabilidad p_m de mutación, definida previamente por el usuario, que es bastante pequeña en la naturaleza, y se mantiene bajo para los AGs, por lo general en el rango de 0.001 y 0.01. De acuerdo con esta probabilidad, el valor del bit se cambia de 0 a 1 o viceversa. De esta manera, un descendiente se produce a partir de un solo padre (Negnevitsky, 2005).



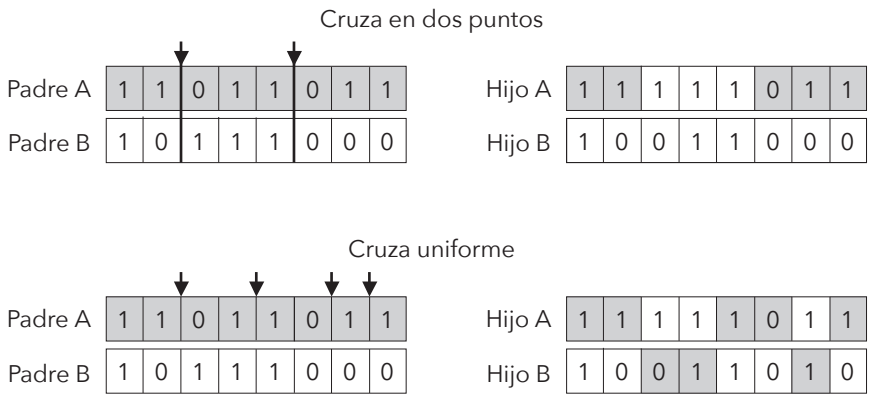


Figura III.4 Operador de cruza.

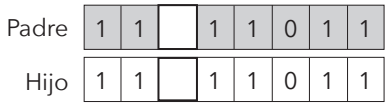


Figura III.5 Operador de mutación.

Los tres operadores siguientes componen el algoritmo CHC (generación cruzada, selección elitista, recombinación heterogénea mediante "prevención de incesto" y mutación cataclismo) y tienen la idea de que la recombinación debe ser el operador de búsqueda dominante.

III.5.2.6 Selección elitista y prevención de incesto

Después de la recombinación, los mejores individuos N se han extraído de la población de los padres y la población descendiente creó la próxima generación; esto implica que los individuos duplicados se eliminan de la población. Esta forma de selección también se conoce como selección de truncamiento (Eshelman, 1991).

Posterior a esta selección, pares de individuos se forman al azar con la nueva población de padres para aplicar la recombinación, formando *N/2* pares de individuos. Sin embargo, el algoritmo CHC también emplea una restricción de recombinación heterogénea como un método de "prevención de incesto" (*incest prevention*). Esto se logra mediante el emparejamiento de aquellos pares de cromosomas que difieren uno del otro en un número de bits, es decir, un umbral de emparejamiento. El

umbral inicial se establece en *L/4*, donde *L* es la longitud de la cadena. Si no se pudiera aplicar alguna recombinación, es decir, si se produce una generación sin descendientes, entonces se reduce el umbral en uno y se vuelve aplicar el proceso nuevamente. Esto sucede porque los cromosomas de la población son muy similares.

III.5.2.7 Operator de cruza HUX (*Half Uniform Crossover*)

El operador de cruza uniforme en la mitad (en inglés, *Half Uniform Crossover*) es un operador donde los bits son intercambiados al azar y de manera independiente: exactamente la mitad de los bits que difieren entre los padres se intercambian, ver figura III.6. El operador HUX (Eshelman, 1991) asegura que la descendencia es equidistante entre los dos padres. Esto sirve como un mecanismo de preservación de la diversidad.

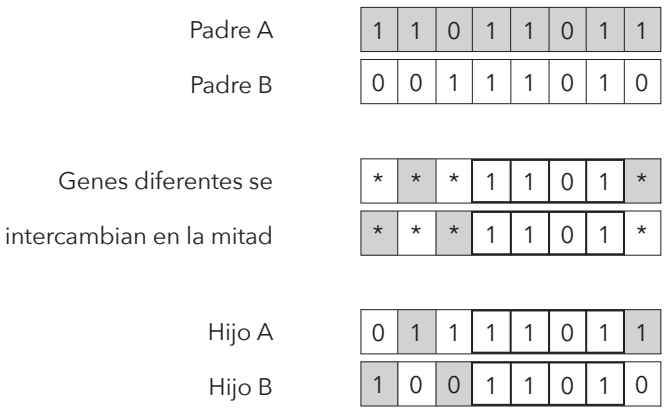
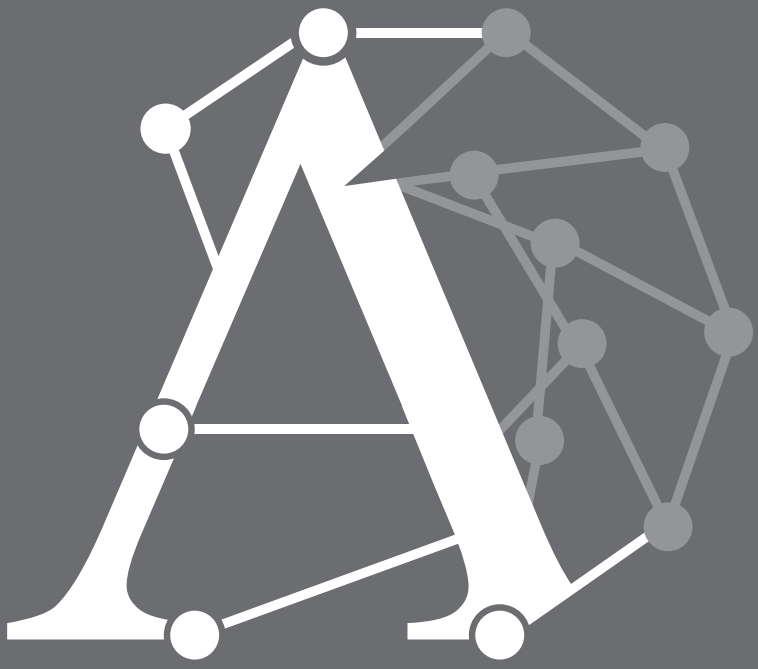


Figura III.6 Representación de la cruza Operador HUX.

III.5.2.8 Mutación cataclismo

No se aplica la mutación durante la fase de búsqueda normal del algoritmo CHC (Eshelman, 1991). Cuando la descendencia no se puede insertar en la población de una generación posterior, y el umbral de acoplamiento ha alcanzado un valor de cero, CHC presenta nueva diversidad en la población a través de una forma de reiniciar. La mutación cataclismo utiliza el mejor individuo de la población como una plantilla para volver a inicializar la población. La nueva población incluye una copia del mejor individuo; el resto de la población se genera mediante la aplicación

de una simple mutación relativamente alta, por ejemplo, 35 % del mejor individuo. El nuevo valor de umbral será el producto de la longitud del cromosoma (*L*) y el porcentaje de la mutación (%) utilizado para generar la nueva población. Hay muchas otras maneras de refrescar la población. Por ejemplo, para rescatar a los mejores individuos *k* y generando el resto al azar, o para rescatar a los mejores individuos *k* y usarlas como plantillas para generar el resto de la población.



CAPÍTULO IV. NUEVO MÉTODO PARA LA GENERACIÓN AUTOMÁTICA DE RESÚMENES DE UN SOLO DOCUMENTO

El objetivo de este capítulo es presentar la propuesta del nuevo método para la GART en un solo documento (sección IV.1), así como un método para calcular la máxima calidad que una herramienta de GART puede alcanzar (sección IV.2). En específico, en la primera sección se describen los métodos de selección de términos, pesado de términos, pesado de oraciones y selección de oraciones, que conforman el nuevo método de GART para un solo documento, con independencia del lenguaje. Cabe mencionar que las definiciones teóricas sobre los elementos del nuevo método propuesto se dieron en el apartado anterior, por lo que en este capítulo sólo se retoman los conceptos.

IV.1 Método basado en SFM y ponderación de grafos para la GART en un solo documento con independencia del lenguaje y del dominio

La presentación del nuevo método seguirá las etapas que normalmente se vienen trabajando para un método de GART.

IV.1.1 Selección de términos

Como se ha visto, en esta etapa se debe definir el término que será extraído del texto para poder representar y cuantificar la importancia de las oraciones. Considerando los términos que pueden ser extraídos del texto de forma independientemente del lenguaje se puede ver que la bolsa de palabras es un sub modelo de los n-gramas cuando n=1. Por lo que, las opciones serían solo entre n-gramas y SFMs. Debido a que las SFMs no limitan su longitud, como lo hacen los n-gramas, consideramos que las SFMs hacen un análisis a través de todos los modelos de n-gramas que se pudieran extraer, por lo que sólo permanecerían aquellos que son tanto frecuentes como maximales.

Nuestra hipótesis es que las SFs pueden expresar las ideas importantes y específicas de un documento. Esto puede ser discutido en términos de *tf-idf* (frecuencia de término–frecuencia inversa del documento,

un concepto bien conocido en la recuperación de información (Baeza & Ribeiro, 1999). Por un lado, la idea expresada por una SF es importante para el documento si aparece en repetidas ocasiones (alta frecuencia de término). Por otro lado, la idea correspondiente debe ser específica para este documento, de lo contrario no existiría en el lenguaje una sola palabra o por lo menos una abreviatura para expresarlo (alta frecuencia inversa de documento).

Un n-grama puede ser parte de otro n-grama más largo. Todas las n-gramas contenidas en una SF también son SFs. Sin embargo, con los argumentos dados anteriormente, se puede derivar que tales n-gramas, más pequeños, pueden no tener ningún significado importante por sí solos. Por ejemplo, "*Estados Unidos de América*" es un nombre propio que representa una entidad, mientras que "*Estados*" o "*Unidos de América*" no lo son. Excepciones al igual que "*Estados Unidos*" no deberían afectar mucho nuestro razonamiento, ya que tienden a ser sinónimo de la expresión más larga y el autor del documento elegiría una u otra forma de referirse a dicha entidad, por lo que no deberían aparecer ambos con frecuencia en el mismo documento.

Es importante tener en cuenta que las SFMs representan de manera compacta al conjunto de SFs, puesto que es posible reproducir a todo el conjunto de SFs a partir de las SFMs, si cada SFM se descompone en todas sus subsecuencias. Esta propiedad debe tenerse en cuenta, puesto que si una oración tiene algunas subsecuencias de una SFM también representa un grado de pertenencia, quizá sea muy decisivo esperar que la SFM deba aparecer tal cual en la oración. Es por eso que más adelante se probarán varios esquemas de sub-términos, todos ellos derivados de las SFMs, con el fin de valorar cuál sería mejor.

Cuando decimos los términos, nos referimos a las características que utilizamos en este paso. Los términos son palabras, n-gramas o SFMs extraídos de un documento. También extraemos términos derivados de SFMs tales como palabras y n-gramas. Detalladamente, se proponen las siguientes variantes de selección de términos:

M: el conjunto de todas las SFMs, es decir, un n-grama $m \in M$ si es una SFM con algún umbral β (se consideran SFMs de 2 o más palabras y $\beta \geq 2$). En el ejemplo de la figura IV.1, M = {*is the most beautiful*}. Además, denotamos por M_2 el conjunto de todas SFMs con $\beta = 2$.

B: bigramas repetitivos, es decir, bigramas con frecuencia de al menos 2. Es fácil demostrar que es el mismo conjunto como el conjunto de todas las bigramas de SFMs: un bigrama $b \in B$ si y sólo si existe un SFM $m \in M$ tal que $b \subseteq m$. Es más, teniendo en cuenta en la última definición M_2 en lugar de M también da el mismo conjunto. En nuestro ejemplo, B = {*is the, the most, most beautiful*}.

W: palabras sueltas (unigramas) de elementos de *B* o, lo que es lo mismo, de *M*. Es decir, una palabra $w \in W$ si existe una bigrama $b \in B$ tal que $w \in b$; es fácil demostrar que $w \in W$ si y sólo si existe un SFM $m \in M$ tal que $w \in m$. Una vez más, teniendo en cuenta M_2 en la última definición también da el mismo conjunto. En nuestro ejemplo, *B* = {*is, the, most, beautiful*}.

N: todos los n-gramas de SFMs, es decir, un n-grama $n \in N$ si existe una SFM $m \in M$ tal que $n \subseteq m$ (incluyendo palabras individuales, es decir, 1-gramas). Una vez más, teniendo en cuenta en la última definición M_2 también da el mismo conjunto, lo que permite el cálculo eficiente del conjunto de *N* en la práctica. En nuestro ejemplo, *N* = {*is, the, most, beautiful, is the, the most, most beautiful, is the most, the most beautiful, is the most beautiful*}. Tenga en cuenta que $W \subset N$, $M \subset N$.

***N* \ *W*, *N* \ M_2, *N* \ (*W* ∪ M_2):** igual que el *N* pero sin incluir 1-gramas, toda la SFM, o ambas; aquí M_2 es el conjunto de SFMs con $\beta = 2$. En nuestro ejemplo, $N \setminus (W \cup M_2)$ = {*is the, the most, most beautiful, is the most, the most beautiful*}.

A. Mona Lisa is the most beautiful picture of Leonardo da Vinci
B. Eiffel tower is the most beautiful tower
C. St. Petersburg is the most beautiful city of Russia
D. The most beautiful church is not located in Europe

Figura IV.1 Ejemplo de 4 oraciones de un texto arbitrario.

Damos las diferentes definiciones de los conjuntos *B* y *W* para mostrar que naturalmente se derivan del concepto de SFM y al mismo tiempo se pueden calcular de manera eficiente.

IV.1.2 Pesado de términos

Se propone un esquema de pesado de SFM que tengan en cuenta la frecuencia de T_i de SFM, la longitud de SFM y la frecuencia de términos derivados de SFM. Los términos T_i pueden ponderarse de diferentes maneras y tener el peso t_i. Este esquema general se define como $p_i(t_j) = X \cdot Y$, donde $p_i(t_j)$ - pesado de término *j* en los documentos *i*, *X* y *Y* se puede determinar como la frecuencia de SFM, la longitud de SFM, y la frecuencia de términos derivados de la SFM. Este esquema de pesado de términos permite detectar cuál de las características de la SFM contribuye mejor para resumir un texto. En concreto, se proponen los siguientes esquemas de pesado de términos:

F. frecuencia del término en SFMs, es decir, el número de veces que el término aparece en el texto dentro de alguna SFM. En el ejemplo de la figura IV.1, *f* (*is*) =3 , ya que se produce 3 veces en el texto dentro de la MFS (*is the most beautiful*).

Si el término en sí es una SFM, entonces esto es sólo la frecuencia de este término en el texto (por ejemplo, para *M*, *f* es el mismo que el peso de término en el Experimento 1, para *W* y *N* no lo es). Bajo ciertas condiciones realistas (SFMs no se cruzan en el texto, las palabras no se repiten dentro de una SFM) *f* es el número de veces que el término aparece en el texto como parte de un bigrama repetitivo. En el ejemplo de la figura IV.1, *f* (*is*) = 3, ya que se produce 3 veces en un bigrama repetitivo "*is the*" (y una vez en contexto que no es repetitivo *church is not*).

L. la longitud máxima de una SFM que contiene el término. En el ejemplo de la figura IV.1, *L*(*is*) = 4, ya que está contenido en un 4-grama SFM *is the most beautiful*.

1. el mismo peso para todos los términos.

IV.1.3 Pesado o ponderación de oraciones

En esta etapa se proponen dos opciones a explorar, una simple que consiste en la suma de la relevancia de los términos y otra más sofisticada que consiste en la ponderación basada en grafos.

IV.1.3.1 Suma de relevancia de los términos

El objetivo de emplear este tipo de ponderación es conocer qué tanto mejora el método propuesto al considerar las SFMs como término, junto con los diferentes pesados disponibles. Para esta etapa, se calcula la suma de los pesos de los términos contenidos en la oración. Cuando una oración S_i tiene peso s_j = sum w_{ij}, la contribución de T_i en S_j es $w_{ij} = f_{ij} \cdot t_i$, donde *f* es una presencia de T_i in D_j, *t* es una importancia de T_i. Aquí *f* es binario.

IV.1.3.2 Ponderación basada en grafos

Una de las principales aportaciones de este método es la propuesta de utilizar el algoritmo TextRank para ponderar las oraciones. Aunque, TextRank ya lo proponía, lo venía haciendo sólo con palabras, en este

caso se propone utilizar las SFMs como nodos del grafo y como aristas se calcularán variantes de los pesados propuestos de forma previa. Finalmente, las oraciones serán ponderadas en la etapa de pesado de oraciones utilizando el algoritmo *PageRank para un grafo no dirigido* (más detalles sobre PageRank en la Sección III.3).

Nodos. Proponemos usar SFMs como vértices de un grafo.

Aristas. Las relaciones que se conectan SFMs son las relaciones de pesado de términos, tales como la frecuencia de SFMs en un texto, la longitud de SFMs y su presencia.

Algoritmo. Utilizamos el algoritmo de rankeo basado en grafos PageRank (la versión para texto de este algoritmo se llama TextRank) para encontrar el rankeo sobre los nodos en el grafo. Iterar el algoritmo de rankeo basado en grafos hasta la convergencia. Ordenar nodos basados en su puntuación final. Utilizar los valores asociados a cada nodo para las decisiones de rankeo/selección.

Algoritmo. Pasos principales:

1. Los vértices del grafo serán cada una de las oraciones, representadas mediante los términos derivados de las SFMs propuestas anteriormente en la sección IV.1.1.
2. Se van a crear aristas entre los vértices en el grafo no dirigido considerando los diferentes pesados propuestos en la sección IV.1.2.
3. Calcular la primera ponderación con base en sus aristas y después aplicar PageRank para ponderar los nodos del grafo para encontrar su pertinencia para un resumen. Iterar el algoritmo de ponderación basado en grafos hasta que converja. Ordenar aristas basadas en su puntuación final. Utilizar los valores asociados a cada arista para las decisiones de ponderación/selección.

En la figura IV.2 se muestra el grafo construido considerando como términos a las SFMs (término nombrado como M en la sección IV.1.1) de las oraciones de la figura III.1. Como pesado de términos se utilizó la variante de peso $(t_k) = f(t_k)^{L(t_k)}$ que significa que se toma como relevancia del término t_k. a la frecuencia de la SFM elevada a la longitud de dicha SFM. El peso inicial de cada arista entre el vérticeV_i y el vertex V_j fue calculado como:

$$\text{Peso Arista } (V_i, V_j) = \prod t_{k=\{t|t\in V_j \cap t\in V_j\}} \text{peso } (t_k) \qquad (4.1)$$

Por ejemplo, considerando las SFMs como términos de las oraciones de la figura III.1 se puede construir el grafo de la figura IV.2, donde para el peso de la arista entre dos oraciones se puede calcular como:

Peso Arista (Vi,Vj). Por ejemplo, la oración *B* tiene las SFMs {*un, patrimonio cultural, fueron*} y la oración *D* tiene las SFMs {*fueron, los faraones, las pirámides de Egipto*}, por lo que su intersección sería {*las pirámides de Egipto*}. Por lo que el PesoArista (B, D) = 2^4=16.

Para calcular la relevancia inicial de cada vértice se suman todas las aristas que tiene ese vértice como se muestra en la siguiente fórmula (4.2):

$$\text{Peso Inicial } (V_i) = \sum_{i=j} \text{peso Arista } (V_i, V_j) \qquad (4.2)$$

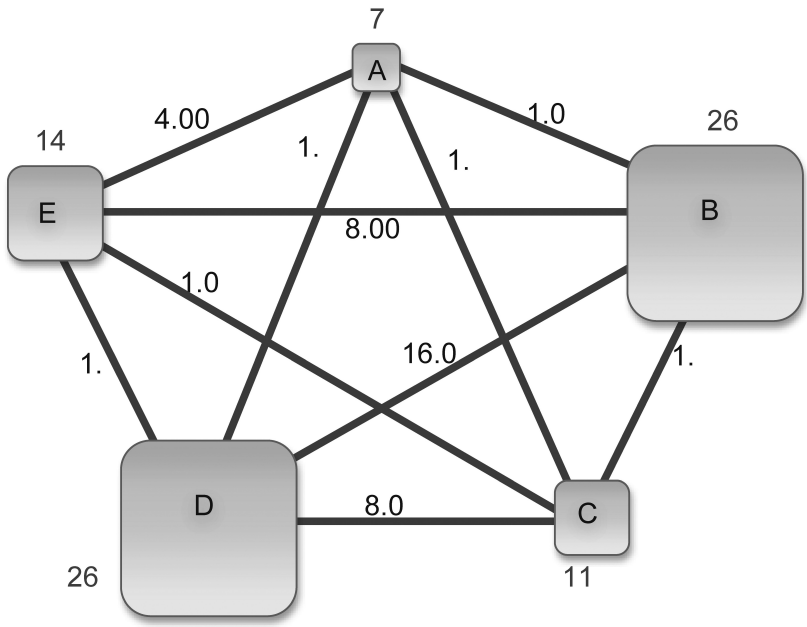


Figura IV.2 Representación del grafo inicial usando SFMs como términos de las oraciones de la figura III.1. El tamaño del nodo representa la importancia inicial de la oración dentro del documento.

Considerando el ejemplo anterior, el vértice *B*=1+8+16+1=26. Como se puede ver en la figura IV.2, los vértices se pueden ordenar de mayor a menor importancia en *B, D, E, C, A*. Si sólo se tuviera que seleccionar una oración como resumen, no habría forma de decidirse entre *B* y *D* puesto que tienen la misma relevancia. Si se tuvieran que seleccionar tres oraciones se seleccionarían *B, D* y *E*.

Sin embargo, la relevancia final de cada oración se obtiene hasta que se haya aplicado el algoritmo de PageRank al grafo. El cual quedaría como el mostrado en la figura IV.3, donde se puede ver que la relevancia de los vértices se modificó ligeramente; queda más claro que la oración *D* es la más importante seguida de *B*. De esta manera, las oraciones quedarían ordenadas de acuerdo a su relevancia en *D, B, E, C, A*.

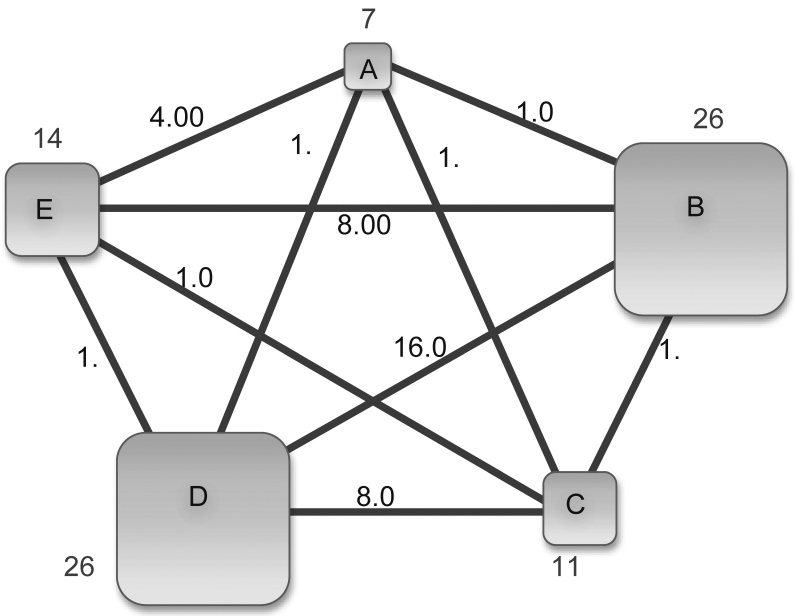


Figura IV.3 Representación del grafo final usando SFMs como términos de las oraciones de la figura III.1. El tamaño del nodo representa la importancia de la oración dentro del documento.

Para medir qué tanto difieren los resultados alcanzados para el ejemplo anterior con los resultados de los encuestados, se puede utilizar la distancia de Manhattan (explicada en la sección III.4.1). La distancia representa la diferencia entre ambos criterios, si fueran idénticos sería de cero. En este caso la combinación de *D, C, B, E, A* dio 5 que es menor a la combinación de *C, D, B, E, A* que dio 6, y la cual es menor al resultado que dio TextRank en su versión original con 10. De esta manera se puede ver que la propuesta de utilizar SFMs concuerda más con el criterio humano al menos en este pequeño ejemplo. Dicho de otra manera, al parecer el algoritmo de TextRank se aleja el doble de la propuesta realizada, no obstante, falta hacer las pruebas correspondientes con varios documentos reales.

Siguiendo con el ejemplo de la encuesta, se les pidió que seleccionaran dos oraciones de la figura III.1 como resumen. Las respuestas ordenadas de acuerdo a los votos recibidos fueron: *D*(7), *B*(6), *C*(3), *E*(2) y *A*(2). Por lo que claramente *D* y *B* tienen la mayor preferencia para ser seleccionadas como resúmenes, lo cual también coincide, y en el mismo orden, con los resultados del método propuesto. Cabe resaltar que el método propuesto señaló como la peor oración a la misma que señalaron los encuestados. De hecho, sólo hay una diferencia entre un voto (con las oraciones *C* y *E*) entre el método propuesto y el criterio de los encuestados.

IV.1.4 Selección de oraciones

Este procedimiento completa un resumen añadiendo las oraciones mejor ponderadas o seleccionando la posición de una oración en un texto hasta que el resumen alcance el número de palabras deseado.

Como la primera opción, se seleccionan las oraciones con mayor peso. Este tipo de métodos es independiente del dominio y se puede aplicar para una variedad de textos. Como la segunda opción, el tipo de métodos es dependiente de la posición y se puede aplicar sólo para temas especiales. Estas dos opciones se resumen como sigue:

– *Best*. Oraciones con mayor peso fueron seleccionadas hasta que se alcance el tamaño deseado del resumen (100 palabras). Este método es el más estándar.

– *K*best+first. Se seleccionan *k* mejores oraciones, y luego se seleccionaron las primeras oraciones del pesado del texto hasta alcanzar el tamaño deseado del resumen. Esto fue motivado por Baseline (se menciona en la sección V.1) que es muy difícil de vencer: sólo las mejores oraciones de acuerdo con nuestro sistema de pesado pueden llegar a estar por encima de Baseline.

IV.2 Nuevo método para calcular el Topline utilizando un algoritmo genético

En esta sección se presenta el método propuesto para encontrar Topline, en otras palabras, la mejor combinación de las oraciones candidatas a formar un resumen. Asimismo, al mejor resultado obtenido para la colección dada le llamamos Topline. Nuestro método se puede aplicar para encontrar Topline no sólo para corpus de generación de resúmenes (por ejemplo, tales como DUC-2001 hasta DUC-2007), sino también para

otras tareas de procesamiento de lenguaje natural. El esquema del método propuesto se muestra en la figura IV.4. En esta sección detallamos el algoritmo genético propuesto.

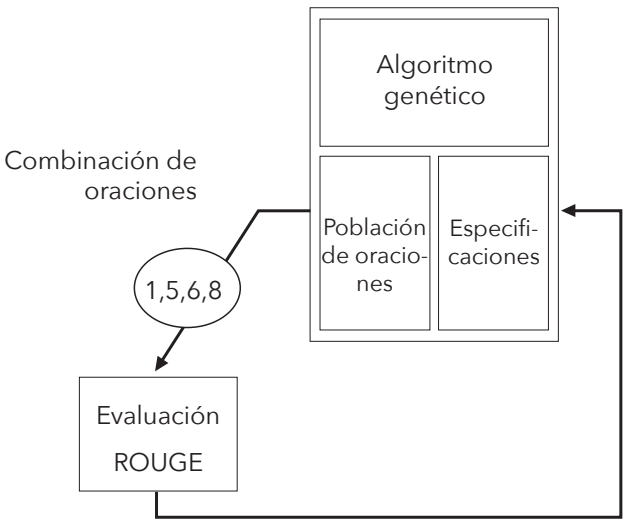


Figura IV.4 Esquema del algoritmo genético propuesto.

El algoritmo genético mantiene una población de cromosomas y cada uno representa una combinación de oraciones candidato. Este algoritmo genético utiliza datos del sistema ROUGE para evaluar la aptitud de cada oración en la población. Esta evaluación se hace en cada paso de tiempo mediante la simulación con cada combinación de las oraciones y la formación de una función de aptitud, basada en la evaluación ROUGE, que caracteriza el rendimiento deseado. Utilizando esta evaluación de la aptitud, el algoritmo genético se propaga el número de oraciones a la siguiente generación a través de la combinación de las operaciones genéticas que se proponen a continuación. La combinación de las oraciones, que es la más apta en la población, se utiliza para componer un resumen.

El procedimiento propuesto para estimar la mejor combinación de oraciones mediante un AG se resume de la siguiente manera (ver figura IV.5):

1. Determinar el número de oraciones del texto dado
2. Construir una población inicial
3. Codificar cada cromosoma en la población
4. Evaluar el valor de la aptitud de cada cromosoma

5. Reproducir cromosomas de acuerdo con el valor de la aptitud calculado en el paso 4
6. Crear descendiente y reemplazar los cromosomas de los padres por los hijos a través de cruce y mutación
7. Volver al paso 3 hasta que se cumpla el número máximo de iteraciones

Representación

Para representar la combinación de oraciones se utiliza cromosomas de longitud $N \cdot B$, donde N es el número de oraciones del texto original y B el número de bits que utilizamos para codificar el número de oraciones.

Población

La población inicial se forma al azar. Su tamaño es fijo e igual a 35 individuos.

Reproducción

Cuando la evaluación está hecha, continuamos con la etapa de reproducción. Consideramos aplicar la estrategia más aproximada a un equilibrio entre la diversidad y la convergencia. Tal estrategia o algoritmo (por ejemplo, algoritmo CHC) implica el uso del operador de reproducción HUX. De esta manera, la nueva población se obtiene aplicando el operador HUX que asegura que los descendientes son equidistantes entre los dos padres. Esto sirve como un mecanismo de preservación de la diversidad.

Función de aptitud

Proponemos la función de aptitud para medir F-measure en cada combinación de oraciones mediante el sistema de evaluación ROUGE. La combinación de oraciones que obtengan la mejor puntuación de *F-measure* será el mejor resumen de un texto.

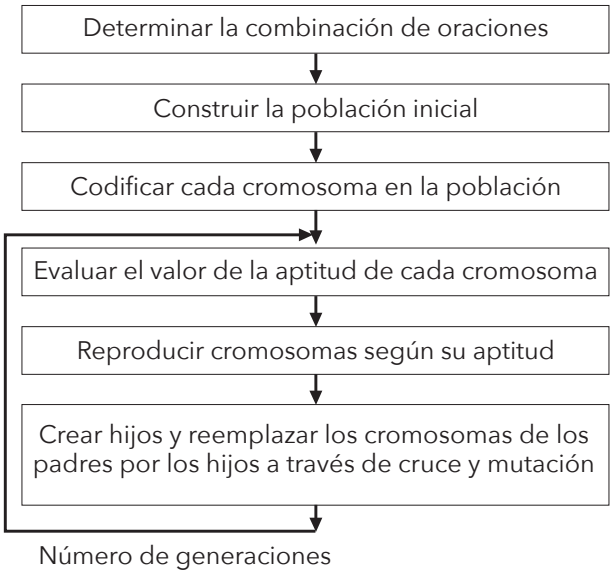
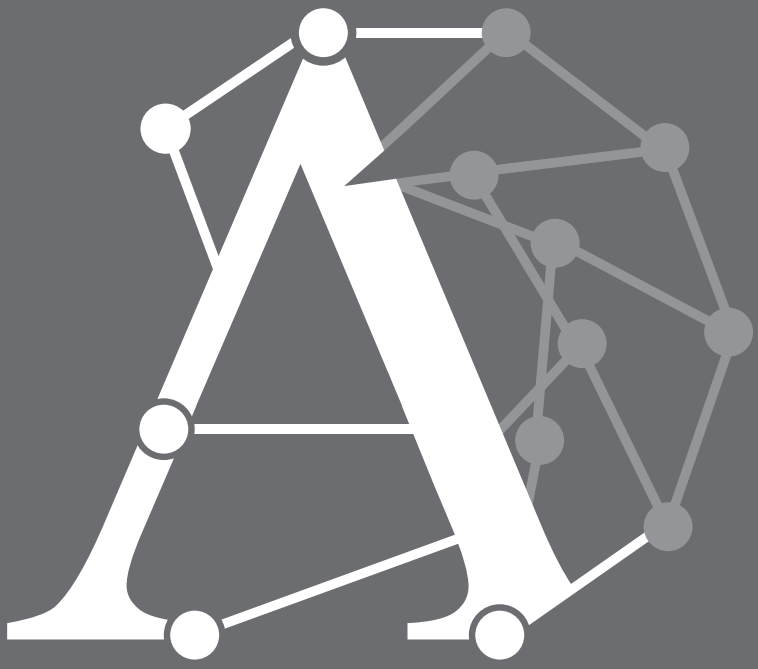


Figura IV.5 Algoritmo genético propuesto.



CAPÍTULO V. RESULTADOS EXPERIMENTALES PARA LA GENERACIÓN AUTOMÁTICA DE RESÚMENES DE UN SOLO DOCUMENTO

En este capítulo se realiza la experimentación con el nuevo método propuesto para la composición de resúmenes de documentos para el corpus DUC-2002 (colección estándar de resúmenes en idioma inglés que se utiliza para comparar los resultados de diferentes métodos de generación de resúmenes de texto). Es decir, se prueban los diferentes esquemas de selección de términos, pesado de términos, pesado de oraciones y la selección de oraciones que se propusieron en el capítulo anterior. Para cada experimento se aplicó el método propuesto y se evaluaron los resúmenes generados. Cada resultado presenta una discusión sobre los porcentajes alcanzados. En la segunda sección se presentan los resultados que se obtuvieron del cálculo del Topline para la colección de documentos DUC2002. Con el cálculo Baseline: aleatorio (la peor calidad que se puede obtener para la GART por un método sin inteligencia) y Topline (la mejor calidad posible que un método pudiera obtener) fue posible recalcular los resultados de todos los métodos y herramientas con el objetivo de observar qué tan significantes son los avances que muestran.

V.1 Elementos para la experimentación

Hemos llevado a cabo varios experimentos para verificar las hipótesis formuladas en los capítulos anteriores.

V.1.1 Algoritmo

En cada experimento se siguió la secuencia estándar de pasos:

Selección de términos. Decidir qué características se van a utilizar para describir las oraciones.

Pesado de términos. Determinar cuál es la importancia de cada característica que se calcula.

Pesado de oraciones. Valorar la importancia de las características que se combinan de acuerdo con la importancia de la oración.

Selección de oraciones. Decidir qué oraciones se seleccionan para el resumen.

Los ajustes específicos para cada paso varían entre los experimentos y se explican en la sección V.2.

V.1.2 Conjunto de datos para prueba

Se utilizó la colección DUC-2002 (DUC, 2014), la cual tiene 567 artículos de noticias de diferentes longitudes y temas. Cada documento de la colección DUC se suministra con un conjunto de resúmenes realizados por dos expertos diferentes. Aunque se pidió a cada experto generar resúmenes de diferente longitud, sólo hemos utilizado las variantes de 100 palabras, por ello se tienen únicamente dos resúmenes por cada noticia de la colección.

V.1.3 Herramienta de evaluación

Se utilizó la herramienta de evaluación ROUGE (Lin, Hovy, 2004). Este sistema tiene la más alta correlación con los juicios humanos (Lin & Hovy, 2003). Compara los resúmenes generados por el programa y los realizados por los humanos (*gold standard*). Para dicha comparación se utilizó la configuración ngram (1, 1) de ROUGE (ver sección II.3), es decir, aquella que guardo una alta correlación con los juicios humanos, llegando a un nivel de 95 % de confianza.

V.1.4. Baseline

Denotamos Baseline: first como la selección de las primeras oraciones del texto original, hasta alcanzar el tamaño deseado (DUC, 2014). Esta configuración da muy buenos resultados en la clase de textos de noticias con las cuales hemos experimentado. También hemos propuesto otra *baseline*, que creemos más realista para aquellos textos diferentes a las noticias. Se llama Baseline: random, y selecciona aleatoriamente oraciones del texto original. Los resultados que se comparten promedian diez corridas de cada experimento (véase los resultados en la tabla V.5).

V.2 Metodología experimental

Ponemos a prueba diferentes configuraciones de selección de términos, pesado de términos, pesado de oraciones y selección de oraciones. Se propone la siguiente metodología experimental:

Experimento 1. Diferentes opciones de selección de términos se ponen a prueba.

Experimento 2. Selección de términos usando SFMs y términos derivados extraídos.

Experimento 3. Selección de términos usando descripciones multipalabra, extraídas de una colección de oraciones, el pesado de términos y la selección de oraciones.

Experimento 4. Selección de términos, pesado de términos y selección de oraciones usando diferentes umbrales.

Experimento 5. Selección de términos, pesado de términos y selección de oraciones con la etapa de preprocesamiento.

Experimento 6. Selección de términos, pesado de términos y selección de oraciones utilizando el algoritmo de grafos.

V.3 Resultados experimentales

De acuerdo con la metodología experimental anteriormente mencionada, en esta sección se presenta con detalle en qué consistió el experimento, los resultados obtenidos y las conclusiones parciales que se pueden obtener a partir de dichos resultados.

V.3.1 Experimento 1

Para la selección del término, comparamos SFMs con características más tradicionales, como palabras individuales y n-gramas. Opcionalmente, las palabras vacías fueron eliminadas en la etapa de preprocesamiento; en este caso nuestros bigramas (o SFMs) podrían incluir más palabras en el texto original, como se explica en el capítulo III. Para el pesado de términos se utilizó la frecuencia del término; para el pesado de oraciones se utilizó la suma de los pesos de los términos contenidos en la oración. En la elección de oraciones se escogieron aquellas con mayor peso hasta alcanzar el tamaño deseado del resumen (100 palabras).

Discusión

Como una revisión de significancia estadística, los datos de prueba se dividieron al azar en dos mitades y se corrió éste, y la mayoría de los otros experimentos, por separado en cada subgrupo. Estas pruebas confirmaron las observaciones cualitativas reportadas en este ensayo.

Como muestra la tabla V.1, las SFMs son una opción prometedora para la selección de términos. Esto motivó nuestros próximos experimentos con esquemas de selección de términos derivados de ellos, así como con las opciones de pesado de términos.

La tabla V.1 muestra los resultados. Las medidas de recuerdo o precisión se pueden utilizar para la comparación, ya que el tamaño de todos los resúmenes es el mismo (100 palabras). Se puede ver un ejemplo del texto completo acompañado por el resumen realizado por el humano y otro por el sistema en el Anexo D.

Tabla V.1 Recuerdo para los resúmenes de 100 palabras para las diferentes opciones de selección de términos

Términos	Con palabras vacías	Sin palabra vacías
W: palabras de B or M	0.39421	0.41371
B: bigramas repetitivos	0.40810	0.42173
M: todas las SFMs	0.43066	0.44085

V.3.2 Experimento 2

Inspirado por los resultados anteriores, se ha experimentado más con SFMs y otras opciones de selección de términos derivados de ellos. Además de M, consideramos una opción *W* de la sección III.

Los resultados se muestran en la tabla V.2. Llevamos a cabo los experimentos en tres fases. De la tabla V.1 sabíamos que el esquema de selección término *M*, con palabras vacías eliminadas, dio los mejores resultados con otros parámetros fijos (pesado de términos, pesado de oraciones y la selección de oraciones). Así que empezamos a partir de la modificación de estos parámetros en el sistema de selección de términos. Véase la tercera parte superior de tabla V.2. La primera línea de la tabla representa el mejor resultado de la tabla V.1. Los mejores resultados se destacan en negritas.

En cada experimento se considera la siguiente configuración del algoritmo principal:

Preprocesamiento. Opcionalmente las palabras vacías fueron eliminadas en la etapa de pre-procesamiento.

Selección de términos. Cada texto original se representa por separado para cada oración. Las SFMs se extraen de cada oración por separado. Las descripciones multipalabras resultantes extraídas de cada oración son diferentes de las descripciones multipalabras extraídas de

un documento completo. Específicamente, la representación de un texto es diferente, con la consecuencia de que las SFMs resultantes son diferentes.

Pesado de términos. Es la frecuencia del término de una SFMs (*f*); la longitud máxima de una SFMs que contiene el término (*l*) y el mismo peso para todos los términos (1).

Pesado de oraciones. Se utilizó la suma de los pesos de los términos contenidos en la oración.

Selección de oraciones. Para formar el resumen se seleccionaron las oraciones con mayor peso hasta alcanzar el tamaño deseado del resumen (*best*). En otro esquema, se seleccionaron las *k* mejores oraciones (*kbest*), y luego se agregaron las primeras oraciones hasta alcanzar el tamaño deseado del resumen (*kbest+first*).

Discusión

Luego intentamos otras opciones de selección de términos, tales como *W*, con la opción 1 de pesado de términos y las opciones relacionadas con *f*, que mostraron un buen desempeño en el primer experimento. Los resultados se muestran en la tercera mitad de la tabla V.2. Selección de términos *W* dio mejor resultado que *M*. Por último, con las mejores combinaciones obtenidas a partir de los dos primeros experimentos, hemos probado diferentes variantes de selección de oraciones; ver la última parte de la tabla V.2.

Se puede observar que cualquier opción de selección de la primera oración del *kbest+first* supera cualquier combinación que utiliza el esquema estándar de selección de oraciones, con menor *k* siempre da mejores resultados; es decir, sólo la más leve corrección a Baseline lo mejora. El mejor resultado se obtuvo con palabras individuales derivadas de SFMs, con su pesado por la frecuencia de la SFM correspondiente.

Tabla V.2 Los resultados del experimento donde las descripciones multipalabras se extraen de cada oración

Términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	l × f	best	0.43734	0.45402	0.44519
	1		0.43881	0.45415	0.44600
	l		0.43824	0.45487	0.44606
	f		**0.44034**	**0.45581**	**0.44759**
	l × l		0.42839	0.44633	0.43685
	l ×× f		0.42588	0.44360	0.43423
W	f	best	**0.44483**	**0.45829**	**0.45134**
	1		0.38367	0.40290	0.39291
W	f	1best+first	**0.46523**	**0.48219**	**0.47344**
		2best+first	0.46214	0.47739	0.46952
M	l	1best+first	0.46306	0.48052	0.47150
	f	1best+first	0.46448	0.48185	0.47288
	1	1best+first	0.46423	0.48143	0.47255

V.3.3 Experimento 3

En cada experimento de esta sección se considera la siguiente configuración del algoritmo principal:

Preprocesamiento. Opcionalmente, las palabras vacías fueron eliminadas en la etapa de pre-procesamiento.

Selección de términos. Cada texto original se representa como una colección de oraciones. Las SFMs se extraen de un documento completo. En este experimento, además de M y W del experimento 2, hemos considerado una opción N y la generalización de los conjuntos de N, N \ W, N \ M2, N \ (W U M2). Donde N \ W es una resta de conjuntos que se lee como el conjunto de N, eliminando los elementos que hay W.

Pesado de términos. Es la frecuencia del término de una SFMs (*f*); la longitud máxima de una SFMs que contiene el término (*l*) y el mismo peso para todos los términos (1).

Pesado de oraciones. Se utilizó la suma de los pesos de los términos contenidos en la oración.

Selección de oraciones. Para formar el resumen se seleccionaron las oraciones con mayor peso hasta alcanzar el tamaño deseado del resumen (*best*). En otro esquema, se seleccionaron las *k* mejores oraciones (*kbest*) y luego se agregaron las primeras oraciones hasta alcanzar el tamaño deseado del resumen (*kbest+first*).

Luego intentamos otras opciones de selección de términos, tales como *W* y *N*, con la opción de ponderación de términos 1 y las opciones relacionadas con f, que mostraron un buen desempeño en el primer experimento. Los resultados se muestran en la tercera mitad de la tabla V.3. La selección de términos *W* dio un resultado ligeramente mejor que *M*. Los resultados para *N* son iguales con f y 1 como ponderación. Otras combinaciones a base de *N* no dieron buenos resultados, ver tabla V.4 (las palabras vacías fueron excluidas y se utilizó *kbest+first*). Por último, hemos intentado diferentes variantes de selección de oraciones, ver el último tercio de la tabla V.3.

Tabla V.3 Resultados para diferentes opciones de detección de términos

Términos	Palabras vacías	Pesado de términos	Selección de oraciones	Recuerdo	Precisión	F-measure
M	excluidos	f	best	0.44085	0.45564	0.44796
		1		**0.44128**	**0.45609**	**0.44840**
		l		0.43977	0.45587	0.44752
		l^2		0.42995	0.44766	0.43847
		l × f		0.43812	0.45411	0.44581
	incluidos			0.43353	0.44737	0.44022
W	incluidos	f	best	0.44582	0.45820	0.45181
	excluidos			**0.44609**	**0.45953**	**0.45259**
		1		0.38364	0.40277	0.39284
		f^2		0.43892	0.45265	0.44556
N		*f* or 1		0.43711	0.45099	0.44383
W	excluidos	f	1best+first	**0.46576**	**0.48278**	**0.47399**
			2best+first	0.46158	0.47682	0.46895
M		1	1best+first	0.46354	0.48072	0.47185
			2best+first	0.46028	0.47567	0.46772
		l	1best+first	0.46381	0.48124	0.47223
			2best+first	0.45790	0.47430	0.46583

Tabla V.4 Resultados para variantes del conjunto N (opciones: excluidos, *best*)

Términos	Pesado de términos	Recuerdo	Precisión	F-measure
N	*f* or 1	0.43711	0.45099	0.44383
	l	0.42911	0.44324	0.43594
N \ W	1	0.42009	0.43693	0.42823
	f	0.41849	0.43532	0.42662
$N \setminus M_2$	1	0.42315	0.43806	0.43035
$N \setminus (W \cup M_2)$		0.41084	0.42759	0.41893

Comparación con el experimento 1

Se puede observar que los resultados para el experimento 2 son mejores que para el experimento 3. Por lo tanto, consideramos la comparación de los resultados de los experimentos 1 y 3:

Estado del arte. Mihalcea autora de los trabajos (Mihalcea & Tarau, 2004), (Mihalcea, 2006) nos proporcionó sus datos, los cuales fueron evaluados en las mismas condiciones que los métodos propuestos. En específico, se evaluó la versión DirectedBackward del algoritmo TextRank (Mihalcea & Tarau, 2004). También presentamos los resultados del algoritmo TextRank original con la implementación del algoritmo PageRank con la versión DirectedBackward del algoritmo TextRank, pero con el procesamiento de datos adicionales para eliminar los datos ruidosos (Mihalcea, 2006), y el algoritmo TextRank con una versión modificada de *PageRank* (Hassan, Mihalcea & Banea, 2007) (ver detalles del procesamiento en Mihalcea & Tarau, 2004; Mihalcea, 2006; Hassan, *et al.*, 2007).

Baseline. utilizamos Baseline: first y Baseline: aleatorio (véase la Sección V.1).

Nuestra propuesta. Comparamos estos métodos con los mejores resultados obtenidos con nuestra propuesta del esquema de selección de oraciones de *best* y 1*best+first*, como se muestra en la tabla V.3. En ambos casos se obtuvieron los mejores resultados con las opciones de *W* sin palabras vacías para la selección de términos y *f* para pesado de términos.

Para la comparación equitativa, separamos los métodos por el tipo de información que utilizan, además del pesado derivado de los términos.

- Ninguno (el texto se considera como una bolsa de oraciones, oración como una bolsa de términos, términos como cadenas)
- Orden de oraciones. Por ejemplo, las primeras oraciones se tratan de forma especial
- Procesamiento sofisticado previo para obtener los términos

Creemos que en futuro la combinación de este tipo de información puede dar mejores resultados. La comparación se presenta en la tabla V.5.

Tabla V.5 Comparación de los resultados de experimento 3 con otros métodos

Información adicional utilizada	Método	Recuerdo	Precisión	F-measure
Ninguna	Baseline: *random*	0.37892	0.39816	0.38817
	TextRank: (Mihalcea, Tarau 2004)	**0.45220**	0.43487	0.44320
	Propuesto: *W*, *f*, *best*	0.44609	**0.45953**	0.45259
Orden de oraciones	Baseline: *first*	0.46407	0.48240	0.47294
	Propuesto: *W*, *f*, 1*best+first*	**0.46576**	**0.48278**	0.47399
Pre-procesamiento	TextRank: (Mihalcea, 2006)	0.46582	0.48382	0.47450
	TextRank: (Hassan, *et al.*, 2007)	**0.47207**	0.48990	0.48068

No pudimos aplicar nuestro método con la opción de preprocesamiento, ya que no se tuvo acceso a los detalles específicos del preprocesamiento utilizado por Mihalcea (2006) y Hassan, *et al.* (2007) (véase experimento 5 para el detalle de pre-procesamiento). Sin embargo, en las otras dos categorías este método superó a los otros. Posiblemente, con el mismo tipo de preprocesamiento nuestro método superaría a los otros también en la última categoría.

Discusión

Observamos que las palabras de los bigramas repetitivos son buenos términos, así sucede con las SFMs (podemos especular que SFMs siguen siendo mejores unidades semánticas, aunque dividiéndolos en palabras individuales da una comparación más flexible y menos dispersa). Para el pesado de términos se observó que un buen sistema de pesado es el número de veces que aparece el término en el texto como parte de un bigrama repetitivo. Con estos ajustes se obtuvieron resultados superiores a los métodos del estado de arte.

La mayoría de los métodos del estado de arte muestran peores desempeños que el método Baseline, ya que éste toma en cuenta el orden especial de las oraciones en las noticias, es decir, un resumen casi listo en sus primeras oraciones. Sin embargo, nuestros métodos pueden seleccionar una oración mejor que el método Baseline (aunque ya la segunda mejor oración seleccionada por nuestro método demuestra ser peor que de Ba-

seline). Esto da un método híbrido (una oración nuestra y luego Baseline) superior a ambos Baseline y otros métodos del estado de arte.

En este experimento no aplicamos pre-procesamiento aunque ha demostrado ser beneficioso para otros métodos, por lo que nuestros resultados son inferiores a aquellos que lo aplican, aunque por encima cuando no lo aplican. Esto último nos hace creer que cuando se aplique el preprocesamiento obtendremos mejores resultados, con respecto a todos los métodos existentes. Este será uno de los experimentos descritos a continuación.

Por otro lado, nuestros experimentos muestran que muy diferentes opciones afectan sólo ligeramente al resultado global, al menos en la colección que utilizamos. Esto probablemente se explica por la naturaleza de los textos de esta colección (informes cortos de noticias) y tal vez por el comportamiento de la herramienta de evaluación ROUGE. Los resultados obtenidos con la configuración de Baseline aleatoria son bastante altos (casi cualquier método daría los resultados similares), mientras aquellos que se esperaba dieran los mejores resultados (selección de las primeras oraciones del texto) arrojaron porcentajes bastante bajos y muy cerca de Baseline: aleatorio.

V.3.4 Experimento 4

Para este ensayo, se utiliza la configuración del algoritmo del experimento 3 (véase el experimento 3). Luego se probó esa configuración con $\beta = 2$, 3, 4 (tabla V.3). Pusimos a prueba los esquemas propuestos con $\beta = 2$ (ver tabla V.6), $\beta = 3$ (ver tabla V.7) y $\beta = 4$ (ver tabla V.8). Los resultados de la comparación se muestran a continuación en las tablas V.9 - V.11.

Tabla V.6 Resultados para experimento 4 con $\beta = 2$

Términos	Palabras vacías	Pesado de términos	Selección de oraciones	Recuerdo	Precisión	F-measure
M	excluidas	l × f	best	0.43731	0.45347	0.44508
		1		**0.43749**	**0.45182**	**0.44438**
		l		0.43731	0.45347	0.44508
		l^2		0.42781	0.44566	0.43640
W	excluidas	f	best	**0.44659**	**0.45968**	**0.45293**
		1		0.38367	0.40290	0.39291
		f^2		0.44114	0.45512	0.44790

W	excluidas	f	1best+first	**0.46536**	**0.48230**	**0.47355**
			2best+first	0.46296	0.47769	0.47009
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46342	0.48069	0.47177
			2best+first	0.45701	0.47320	0.46484

Tabla V.7 Resultados para experimento 4 con $\beta = 3$

Términos	Palabras vacías	Pesado de términos	Selección de oraciones	Recuerdo	Precisión	F-measure
M	excluidas	l × f		0.43470	0.45120	0.44247
		1		**0.43701**	**0.45310**	**0.44459**
		l		0.43470	0.45120	0.44247
		l^2		0.42686	0.44463	0.43525
W	excluidas	f	best	**0.44397**	**0.45773**	**0.45062**
		1		0.38367	0.40290	0.39291
		f^2		0.43797	0.45220	0.44485
W	excluidas	f	1best+first	**0.46622**	**0.48407**	**0.47486**
			2best+first	0.46223	0.47806	0.46989
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46631	0.48392	0.47483
			2best+first	0.46007	0.47638	0.46796

Tabla V.8 Resultados para experimento 4 con $\beta = 4$

Términos	Palabras vacías	Pesado de términos	Selección de oraciones	Recuerdo	Precisión	F-measure
M	excluidas	l × f		0.43013	0.44680	0.43812
		1		**0.43266**	**0.44861**	**0.44025**
		l		0.43013	0.44680	0.43812
		l^2		0.42354	0.44084	0.43183
W	excluidas	f	best	**0.44631**	**0.46505**	**0.45536**
		1		0.38367	0.40290	0.39291
		f^2		0.43712	0.45138	0.44402
W	excluidas	f	1best+first	**0.46788**	**0.48537**	**0.47634**
			2best+first	0.46397	0.47985	0.47165
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46568	0.48373	0.47441
			2best+first	0.45977	0.47604	0.46734

Comparación con el experimento 3

En este experimento se obtuvieron mejores resultados con esquemas propuestos utilizando diferentes umbrales. Aquí, comparamos los mejo-

res resultados del experimento actual (ver tablas V.9 - V.11). Detectamos que la mejor configuración para SFMs como términos seleccionados se obtuvo con la combinación de umbral (β = 2, 3, 4). También, detectamos que para los términos derivados de SFMs el mejor umbral es β = 2. Los resultados de la configuración con combinación de oraciones y con β = 4 es el mejor resultado obtenido.

Tabla V.9 Resultados con términos de SFMs y diferentes umbrales

Método	Recuerdo	Recuerdo	F-measure
M donde β = 2, 3, 4	**0.44128**	**0.45609**	**0.44840**
M donde β= 2	0.43749	0.45182	0.44438
M donde β = 3	0.43701	0.45310	0.44459
M donde β = 4	0.43266	0.44861	0.44025

Tabla V.10 Resultados con términos derivados de SFMs y diferentes umbrales

Método	Recuerdo	Recuerdo	F-measure
M donde β = 2, 3, 4	0.44582	0.45820	0.45181
M donde β = 2	**0.44659**	**0.45968**	**0.45293**
M donde β = 3	0.44397	0.45773	0.45062
M donde β = 4	0.44090	0.45509	0.44776

Tabla V.11 Resultados con combinación de oraciones y diferentes umbrales

Método	Recuerdo	Precisión	F-measure
M donde β = 2, 3, 4	0.46576	0.48278	0.47399
M donde β = 2	0.46536	0.48230	0.47355
M donde β = 3	0.46622	0.48407	0.47486
M donde β = 4	**0.46788**	**0.48537**	**0.47634**

Discusión

Sólo hay cinco mejores sistemas (Mihalcea, 2006) que la configuración Baseline con pequeñas diferencias en los resultados. En el experimento anterior se obtuvieron mejores resultados que la *baseline*. Para la colección de DUC2002 los resultados de la configuración de la *baseline* son muy altos debido a que la mayoría de los textos consisten en las descripciones de noticias y en este tipo de textos es común que las primeras oraciones describan, brevemente, la noticia dada. En otras palabras, algunas de las primeras oraciones son resúmenes de un texto dado. De ahí

que en otros tipos de textos la configuración de *baseline* no funcionaría. Por lo tanto, es justo comparar con los métodos del estado de arte como *caminos aleatorios* (Mihalcea, 2006). El autor de este trabajo proporcionó los datos de sus resúmenes que fueron evaluados en las mismas condiciones que los métodos propuestos. En específico se evaluó la versión DirectedBackward de TextRank (ver tabla V.12, TextRank). Por último, se incluyen los mejores resultados de los métodos propuestos.

Tabla V.12 Comparación de resultados de los experimentos 2 y 3 con otros métodos

Información adicional utilizada	Método	Recuerdo	Precisión	F-measure
Ninguna	Baseline: random	0.37892	0.39816	0.38817
	TextRank: (Mihalcea, Tarau, 2004)	**0.45220**	0.43487	0.44320
	Propuesto: *Z, best*	0.44659	**0.45968**	**0.45293**
Orden de oraciones	Baseline: first	0.46407	0.48240	0.47294
	Propuesto: *Z,* 1*best+first*	**0.46788**	**0.48537**	**0.47634**

Se probaron diferentes combinaciones de selección de términos, pesado de términos, pesado de oraciones y esquemas de selección de oraciones con diferentes umbrales. Con el primer experimento, se observó que las SFMs son buenos términos y nos ayudan a obtener buenos resultados que se comparan con las palabras y n-gramas. En el segundo experimento, se probaron los esquemas propuestos con distintos umbrales. Llegamos a la conclusión de que las palabras derivadas de las SFMs son los mejores términos con $\beta = 2$ y las SFMs son buenos términos con $\beta = 2, 3, 4$.

V.3.5 Experimento 5

Los resultados del experimento se presentan en la tabla V.13 donde se realiza pre-procesamiento, excluyendo las palabras vacías. Los mejores resultados se destacan con letra negrita. Detectamos que el sistema de pesado de frecuencia de palabras, derivadas de SFMs, ofrece la mejor oración del resumen, que combinada con las oraciones obtenidas por la configuración de Baseline dan el mejor resumen. Para la primera parte de este experimento se excluyeron las palabras vacías.

Tabla V.13 Resultados de configuración del experimento 2 usando preprocesamiento (palabras vacías excluidas)

Selección de términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	l × f	best	0.42689	0.43347	0.43005
	1		0.44193	0.44426	0.44298
	l		0.42263	0.42961	0.42599
	f		**0.44678**	**0.44849**	**0.44752**
W	f	best	**0.45504**	**0.45626**	**0.45553**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46416	0.48090	0.47226
		2best+first	0.46033	0.47532	0.46759
M	1	1best+first	**0.46266**	**0.47979**	**0.47094**
	f	1best+first	0.44605	0.44771	0.44676

Para la segunda parte de este experimento cambiamos la configuración del preprocesamiento: a las SFMs se aplica el proceso de *stemming* y las palabras vacías se excluyeron de las SFMs. Ver resultados en tabla V.14.

Tabla V.14 Resultados de configuración del experimento 2 usando preprocesamiento (*stemming* y palabras vacías excluidas)

Selección de términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	l × f	best	0.42538	0.43151	0.42831
	1		0.44315	0.44517	0.44405
	l		0.41837	0.42496	0.42153
	f		**0.44538**	**0.44681**	**0.44598**
W	f	best	**0.45576**	**0.45679**	**0.45615**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46413	0.48081	0.47220
		2best+first	0.46259	0.47721	0.46966
M	1	1best+first	**0.46456**	**0.48169**	**0.47285**
	f	1best+first	0.46432	0.48139	0.47258

Para la tercera parte de este experimento se aplicó el proceso de *stemming* a las SFMs y se incluyeron las palabras vacías. Los resultados se muestran en la tabla V.15.

Comparación con el experimento 3

Comparamos con los métodos del estado de arte como TextRank (Mihalcea, 2006). En específico, la versión DirectedBackward de TextRank fue evaluada en las mismas condiciones que los métodos propuestos (ver tabla V.16, TextRank) y la misma versión de TextRank con preprocesamiento (ver tabla V.16, TextRank con preprocesamiento). También se comparan los resultados presentados en el experimento 3 (ver tabla V.16, SFM sin preprocesamiento). Por último, se incluye la mejor versión de cada experimento (ver SFM con preprocesamiento 1, 2, y 3).

Tabla V.15 Resultado de configuración del experimento 2 usando preprocesamiento (*stemming* y palabras vacías incluidas)

Selección de términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	l × f	best	0.43386	0.43673	0.43494
	1		0.43971	0.44234	0.44067
	l		0.43380	0.43664	0.43487
	f		**0.43867**	**0.44100**	**0.43949**
W	f	best	**0.44609**	**0.44632**	**0.44608**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46486	0.48189	0.47310
		2best+first	0.46293	0.47831	0.47037
M	1	1best+first	0.46461	0.48182	0.47293
	f	1best+first	**0.46508**	**0.48233**	**0.47343**

Podemos ver que el preprocesamiento no afecta positivamente la obtención de términos para el resumen de extracción, por lo menos no en el caso de las SFMs.

Discusión

Hemos modificado nuestro método automático para generación de resúmenes de texto de un solo documento automático, basado en las SFMs como términos, mediante la inclusión de la etapa de preprocesamiento. Sin embargo, el preprocesamiento no afecta positivamente los resúmenes obtenidos con nuestro método. Estas son buenas y malas noticias. Malas porque no hemos encontrado mejores condiciones y nuestros re-

súmenes no mejoraron. Buenas porque se confirmó que las SFMs clásicas (secuencias de formas de palabras y sin aplicación del proceso de *stemming* o solamente palabras importantes), que se calculan de forma totalmente independiente del lenguaje, son buenas condiciones para esta tarea, haciendo nuestro método más robusto.

Tabla V.16 Comparación del resultado de preprocesamiento con otros métodos

Método	Recuerdo	Precisión	F-measure
TextRank	0.45220	0.43487	0.44320
TextRank con pre-procesamiento	0.46582	0.48382	0.47450
MFS sin pre-procesamiento	0.46576	0.48278	0.47399
MFS con pre-procesamiento 1	0.46266	0.47979	0.47094
MFS con pre-procesamiento 2	0.46456	0.48169	0.47285
MFS con pre-procesamiento 3	0.46508	0.48233	0.47343

Por otro lado, ya que se mostró que el pre-procesamiento casi no afecta negativamente a los resultados, entonces se pueden excluir las palabras vacías y el *stemming* del pre-procesamiento, y aun así obtener casi la misma calidad de los resúmenes extractivos. Excluir las palabras vacías reduce significativamente el riesgo del crecimiento exponencial del tamaño de las estructuras de datos utilizadas para extraer las SFMs y para la aplicación de nuestro método, así como el número de los términos (SFMs o n-gramas) para tratar.

V.3.6. Experimento 6

Una vez que se ha probado que las SFMs, como términos multipalabra, junto con las combinaciones de los pesados de frecuencia y longitud, representan la relevancia de dicho término, se probará qué tanto mejoran los resultados al utilizar el algoritmo de grafos de TextRank para pesar las oraciones. En cada experimento, se considera la siguiente configuración del método propuesto:

Selección de términos. *M, W.*

Pesado de términos. La frecuencia del término en las SFMs (*f*); la longitud máxima de la SFM que contiene el término (*l*) y el mismo peso para todos los términos (1).

Pesado de oraciones. Se utilizará PageRank

Selección de oraciones. Las oraciones con mayor peso fueron seleccionadas hasta obtener el tamaño de síntesis deseado.

Los resultados se muestran en las tablas V.17 y V.18. El tamaño de los resúmenes es de 100 palabras. Para la comparación se utiliza la medida F-measure. Los mejores resultados se destacan en negritas.

En la tabla V.17 se utiliza la normalización. Cuando se calcula el peso de las oraciones éste se divide entre el número de palabras.

Tabla V.17 Resultados de algoritmo de grafos (se utilizó la normalización)

Selección de términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	f	best	0.48009	0.47757	0.47865
	f^2		0.48056	0.47801	0.47910
	1		0.46668	0.48337	0.47474
	l		0.48025	0.47773	0.47881
	l^2		**0.48058**	**0.47812**	**0.47917**
	f × l		0.48060	0.47810	0.47916
	f ×× l		0.48079	0.47831	0.47937
W	f	best	**0.48659**	**0.48324**	**0.48473**
	1		0.47682	0.47604	0.47626
	f^2		0.48705	0.48235	0.48451
W	f	1best+first	0.47603	0.47518	0.47543
		2best+first	0.47718	0.47621	0.47652
M	l	1best+first	0.47783	0.47699	0.47724
		2best+first	0.48212	0.48088	0.48132
	f	1best+first	0.47797	0.47712	0.47737
		2best+first	**0.48211**	**0.48093**	**0.48134**

Llevamos a cabo nuestros experimentos en tres fases. A partir de los resultados de otros métodos, sabíamos que el esquema de selección de términos *M,* con palabras vacías eliminadas, dio los mejores resultados con otros parámetros fijos (pesado de términos, pesado de oraciones, y selección de oraciones). Así, se ha empezado a modificar estos parámetros en el esquema de selección de términos, ver la parte superior de la tabla V.17.

Tabla V.18 Resultados de algoritmo de grafos

Selección de términos	Pesado de términos	Selección de oraciones	Resultados		
			Recuerdo	Precisión	F-measure
M	f	best	0.48803	0.48533	0.48626
	f^2		0.48746	0.48482	0.48572
	1		0.47484	0.49180	0.48283
	l		**0.48823**	**0.48577**	**0.48658**
	l^2		0.48741	0.48518	0.48587
	f × l		0.48796	0.48529	0.48620
	f ×× l		0.48716	0.48497	0.48564
W	f	best	**0.48821**	**0.48424**	**0.48604**
	1		0.47529	0.47483	0.47489
	f^2		0.48784	0.48322	0.48534
W	f	1best+first	0.47694	0.47612	0.47635
		2best+first	0.47870	0.47761	0.47798
M	l	1best+first	0.47711	0.47623	0.47650
		2best+first	0.48064	0.47923	0.47976
	f	1best+first	0.47738	0.47649	0.47676
		2best+first	**0.48148**	**0.48016**	**0.48065**

Posteriormente, intentamos otras opciones de selección de términos, tales como *W*, con la opción de pesado de términos 1 y las opciones relacionadas con *f*, que mostró el mejor desempeño en el primer experimento. Los resultados se muestran en la tercera mitad de la tabla V.17. Para la selección de términos, *W* dio un resultado mucho mejor que *M*. Descartamos reportar la selección de términos *N*, ya que los resultados obtenidos no fueron mejores. Otras combinaciones basadas en *M* y *W* no dieron buenos resultados en comparación con otro método en el que esta opción le dio los mejores resultados, ver la parte de abajo de la tabla V.17.

Finalmente, se descartó la normalización de pesado de términos y podríamos obtener mejores resultados para la selección de términos *M* y *W*, donde *M* es un poco mejor que *W* (ver tabla V.18). Se puede observar que cualquier opción de selección de oraciones *kbest+first* no supera el esquema de selección de oraciones estándar. El mejor resultado se obtuvo con las SFMs, con su pesado por la longitud de la SFM correspondiente.

En la figura V.1 se muestra la comparación de los métodos y herramientas del estado del arte con los tres mejores resultados del método propuesto. Como se puede ver, el primer esquema (SFMs: k-best) mejora

los resultados de TextRank, pero no de la mejor herramienta comercial. El segundo esquema (SFM: 1best+first) logra mejorar a las herramientas comerciales. Sin embargo, este último esquema, aunque es independiente del lenguaje, todavía es más dependiente del dominio de noticias ya que combina las primeras oraciones. Por último, la propuesta de las SFMs con grafos sin pre-procesamiento se muestra de manera superior a los trabajos anteriores, con la ventaja de ser independiente del dominio y del lenguaje.

En la gráfica de la figura V.1, la línea punteada representa el cálculo de lo que el peor método, uno que sólo seleccione oraciones al azar, puede obtener como la línea base de fondo (Baseline: aleatorio). Esto genera una pregunta, ¿cuál sería la línea base superior (Topline) que todo método desea alcanzar?, o aún mejor ¿cómo poder calcular el Topline? En la siguiente sección se presenta la propuesta de un algoritmo genético para calcular el Topline.

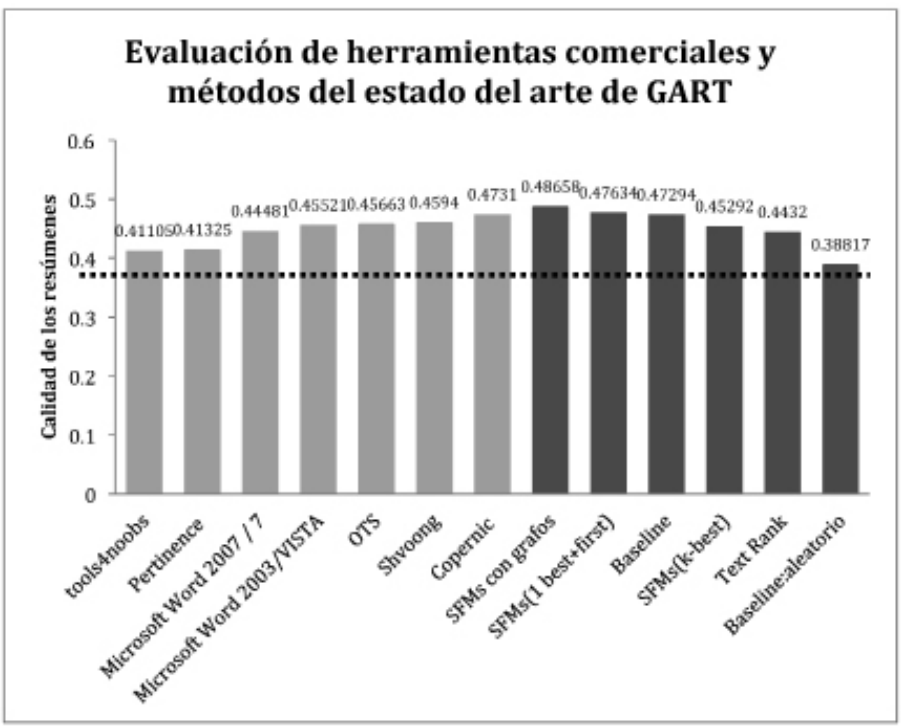


Figura V.1 Comparación de los métodos y herramientas del estado del arte con los mejores resultados del método propuesto para la GART.

V.3.7 Cálculo del Topline usando algoritmos genéticos

Considerando las 567 noticias de la colección de documentos DUC-2002, se ordenaron todos los documentos basándose en el número de oraciones. Los primeros documentos de la lista son documentos cortos con un número pequeño de oraciones, al final de la relación se encuentran aquellos con más frases. Este listado sirve para evaluar mediante ROUGE cada una de las combinaciones de oraciones que pueden formar un resumen. Esto se hace con el fin de saber qué combinación da los mejores resultados según ROUGE con F-*measure*. Debido a que el número de combinaciones crece de manera exponencial, según el número de oraciones de entrada, es prácticamente imposible (por el tiempo que lleva) calcular el Topline de los documentos más grandes. Por lo anterior, se propone utilizar un algoritmo genético para generar una población de posibles soluciones, mismas que evolucionarán buscando una combinación que permita optimizar la función de aptitud, en este caso es ROUGE con F-*measure*.

En la tabla V.19 se muestran los resultados del Topline, calculado a partir de la generación de todas las combinaciones posibles, para los primeros 400 documentos, donde el promedio sería de 0.6297.

Tabla V.19 Resultados de Topline probando todas las combinaciones de oraciones

Número de oraciones	F-measure
1-49	0.67297
50-99	0.65268
100-149	0.63767
150-199	0.62785
200-249	0.61697
250-299	0.59601
300-400	0.60715
Total	**0.62971**

Posteriormente, se hizo el cálculo del Topline para toda la colección el cual se muestra en la tabla V.20. Como puede verse en las tablas V.19 y V.20, la diferencia entre ambos resultados para los primeros 400 documentos no es significativa, lo que valida, en cierta forma, que los resultados alcanzados por el AG son buenos. Esto permite confiar en los resultados que alcanzará el AG para el resto de documentos donde no fue posible calcular todas las combinaciones.

Tabla V.20 Cálculo del Topline utilizando el AG propuesto

Número de oraciones	F-measure
1-49	0.6720
50-99	0.6514
100-149	0.6346
150-199	0.6218
200-249	0.6095
250-299	0.5821
300-349	0.5824
350-399	0.5841
400-449	0.5578
450-499	0.5553
500-549	0.5408
550-568	0.5250
Total	**0.5931**

En la tabla V.21 se muestran los resultados finales donde se combinaron los resultados de las tablas V.19 y V.20 a partir de los cuales fue posible calcular el Topline para la colección de documentos DUC-2002. El promedio fue de **0.596.**

Tabla V.21 Resultados finales de Topline considerando todas las combinaciones de oraciones (0-299) y el algoritmo genético propuesto (300-368)

Número de oraciones	F-measure
1-49	0.67297
50-99	0.65268
100-149	0.63767
150-199	0.62785
200-249	0.61697
250-299	0.59601
300-349	0.5824
350-399	0.5841
400-449	0.5578
450-499	0.5553
500-549	0.5408
550-568	0.5250
Total	**0.5960**

Con los resultados obtenidos en esta investigación, Topline y Baseline: aleatorio es posible recalcular los resultados para ver qué tan significativos son los datos alcanzados por cada trabajo, pues de acuerdo con la figura V.1 medio punto porcentual es significante. Dicho de otra manera, según los resultados de la figura V.1 pareciera que se hizo mucho trabajo sólo para mejorar de medio a un punto porcentual. Para recalcular los datos se consideró a Baseline: aleatorio como el 0 % y el Topline como el 100 %.

Como se puede ver en la figura V.2, el avance alcanzado es más significativo. Se puede ver que *SFMs* con grafos es 19.9 % mejor que TextRank. También cabe señalar que Copernic, aunque dio buenos resultados, es una herramienta que utiliza conocimiento dependiente del lenguaje.

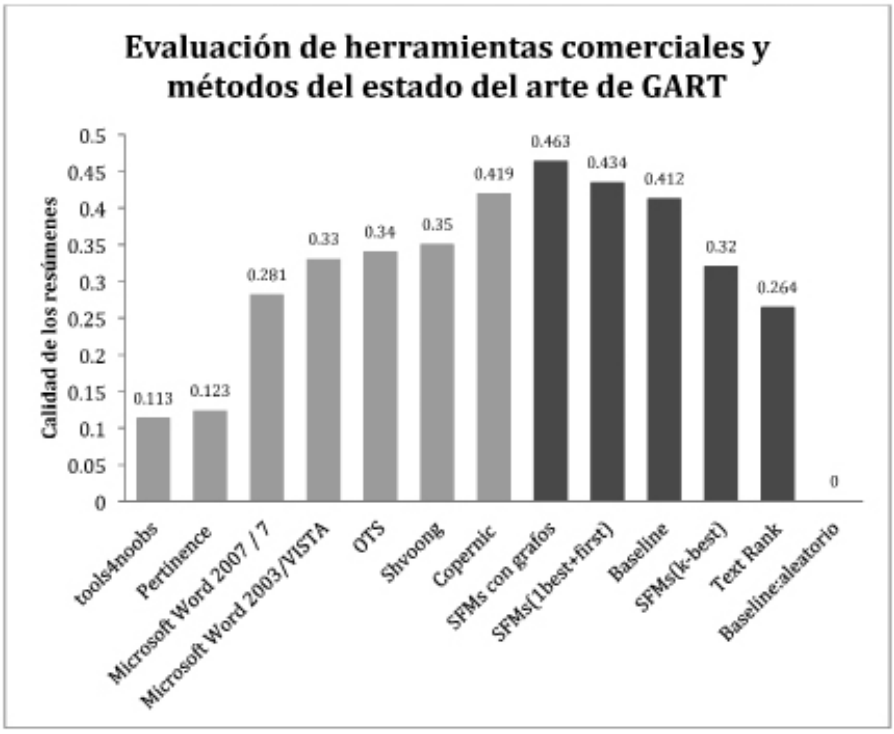


Figura V.2 Avance significativo de los métodos, herramientas del estado del arte y los mejores resultados del método propuesto para la GART.

CAPÍTULO VI.
CONCLUSIONES

En este libro se presentó un panorama del procesamiento del lenguaje natural, enfocado al tratamiento automático de texto en la tarea de generación automática de resúmenes de texto (GART). En específico, se trató el problema de la GART de un solo documento, con el objetivo de analizar las posibles tecnologías que, en cada una de las etapas de la GART, permitieron proponer un nuevo método con un doble reto: mejorar la calidad de los resúmenes y depender lo menos posible de lenguaje y del dominio. El desarrollo de estos métodos para la generación automática de resúmenes de texto nos permite contribuir de manera eficiente al área de procesamiento del lenguaje natural.

En específico, el método propuesto incluye la descripción de las etapas de selección de términos, la ponderación de términos, pesado de oraciones y selección de oraciones. Para cada experimentó se presentó la configuración de los métodos propuestos y los resultados correspondientes. Asimismo, la discusión de los resultados experimentales, la comparación entre diferentes experimentos de este libro y el estado de arte se especificaron explícitamente.

En particular, se alcanzaron los siguientes objetivos:

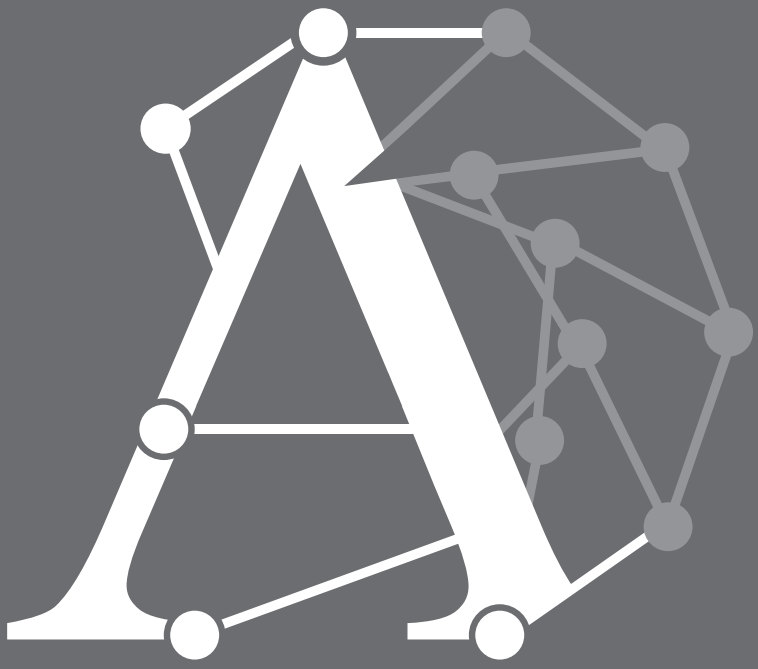
- Identificación de cuatro etapas generales que sigue un método de GART extractivo.
- Evaluación de la calidad de los resúmenes generados por una gran cantidad de herramientas comerciales y métodos del estado del

arte para la GART en un solo documento, lo que establece un estado del arte más real sobre esta área de investigación.

- El cálculo del Topline en la GART de tipo extractivo para un documento permitió saber, por un lado, que los resultados del método propuesto mejoran significativamente con respecto a otros métodos y herramientas comerciales. Por el otro lado, el cálculo del Topline posibilitó saber que también hay mucho por hacer en esta área de investigación.
- Un nuevo método para la generación automática de resúmenes de texto extractivos de un documento, de forma independiente de lenguaje y del dominio, con resultados superiores a los del estado del arte y a las herramientas comerciales.
- Para la etapa de selección de términos se propuso utilizar las secuencias frecuentes maximales como un término independiente del lenguaje y del domino, el cual enriquece su representación por las descripciones multipalabra que contiene.
- En el pesado de términos se propusieron varios esquemas, básicamente se propone el uso de la longitud del término como una métrica para ponderar su relevancia, y a partir de ella generar otras combinaciones.
- Para la etapa de pesado y selección de oraciones se propuso utilizar el algoritmo de PageRank para ponderar la relevancia de cada oración de acuerdo a las oraciones que contiene el documento.

El método propuesto fue probado con el *corpus* DUC-2002. Esta es una colección estándar de resúmenes en idioma inglés, propuesta en la conferencia sobre resúmenes de texto, que facilita la comparación de los resultados obtenidos por los investigadores del área de generación de resúmenes de texto. Sin embargo, aunque el método propuesto, por su forma de trabajo, es independiente del lenguaje y del dominio, sería recomendable, más adelante, probar este método en otros dominios y lenguajes para saber si la calidad de los resúmenes obtenidos también es adecuada.

En respuesta a la pregunta de investigación de este trabajo, podemos decir que las partes más importantes del texto se pueden detectar automáticamente utilizando las secuencias frecuentes maximales como descripciones multipalabras que enriquecen los términos extraídos.



REFERENCIAS

Aceves, R. M., Montes, M., & Villaseñor, L. (2007). Enhancing cross-language question answering by combining multiple question translations. En A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing. 8th International Conference*, CICLing 2007 (pp. 485-493). Mexico: Springer.

Ahonen, H. (1999). Finding All Maximal Frequent Sequences in Text. En D. Mladenic & M. Grobelnik (Eds.), *Proceedings of the 16th International Conference on Machine Learning* (ICML-99), (pp. 11–17). Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?-doi=10.1.1.43.2684&rep=rep1&type=pdf

Ahonen, H. (1999b). Knowledge Discovery in Documents by Extracting Frequent Word Sequences. *Library Trends*, 48(1), 160–181.

Ahonen, H. (2002). Discovery of Frequent Word Sequences in Text. En D. J. Handan, N. M. Adams & R. J. Colvot (Eds), *Pattern Detection and Discovery ESF Exploratory Workshop London, UK, September 2002 Proceedings* (pp. 180–189). London: Springer-Verlag.

Ahonen, H., Heinonen, O., Klemettinen, M., & Verkamo, A. I. (1999a). Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery. En R. Feldman. (Ed.), *Proceedings of 16th International Joint Conference on Artificial Intelligence* (IJCAI-99),

(pp. 1–9). Recuperado de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4013

AMPLN. Asociación Mexicana para el Procesamiento del Lenguaje Natural. (2014). *¿Qué es Procesamiento del Lenguaje Natural?* AMPLN. Recuperado de http://www.ampln.org/pmwiki.php?n=Main.PLN

Baeza, R., & Ribeiro, B. (1999). *Modern Information Retrieval*. Boston: Addison-Wesley Longman Publishing Co. Inc.

Barzilay, R. (2003). *Information fusion for multidocument summarization: paraphrasing and generation* (Tesis doctoral). Columbia University, New York.

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. En I. Mani, M.T. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 111–121). Cambridge, MA: MIT Press.

Barzilay, R., & McKeown, K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3). Recuperado de http://www.mitpressjournals.org/doi/pdf/10.1162/089120105774321091

Bolshakov, I. A. (2004b). Getting one's first million... collocations. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 229-242). Recuperado de http://link.springer.com/chapter/10.1007%2F978-3-540-24630-5_28

Bolshakov, I. A., Bolshakova, E. I., Kotlyarov, A. P., & Gelbukh, A. (2008). Various criteria of collocation cohesion in internet: Comparison of resolving power. En A. Gelbukh, *International Conference on Intelligent Text Processing and Computational Linguistics*, vol. 4919 (pp. 64-72). Recuperado de http://link.springer.com/chapter/10.1007/978-3-540-78135-6_6

Bolshakov, I. A., Galicia, S. N., & Gelbukh, A. (2005). Detection and Correction of Malapropisms in Spanish by means of Internet Search. En V. Matoušek, P. Mautner y T. Pavelka (Eds.), *International Conference on Text, Speech and Dialogue* (pp. 115-122). Recuperado de http://link.springer.com/chapter/10.1007/11551874_15

Bolshakov, I., & Gelbukh, A. (2000). A very large database of collocations and semantic links. En M. Bourzeghoub, *et al.* (Eds.), *International*

Conference on Application of Natural Language to Information Systems (pp. 103-114). Recuperado de http://link.springer.com/chapter/10.1007/3-540-45399-7_9

Bolshakov, I. A., & Gelbukh, A. (2004a). *Computational linguistics models, resources, applications*. Recuperado de http://www.gelbukh.com/clbook/Computational-Linguistics.pdf

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18), 3825-3833.

Brunn, M., Chali, Y., & Pinchak, C. J. (2001). Text Summarization Using Lexical Chains. En *Proc. of Document Understanding Conference 2001*. Recuperado de http://duc.nist.gov/pubs.html#2001.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 335-336). Recuperado de http://dl.acm.org/citation.cfm?id=291025

Carlson, L., Marcu, D., & Okurowski, M. E. (2003). Building a discourse-tagged corpus in the framework of rhetorical structure theory. En J. Kuppevelt & R. Smith (Eds), In *Current and new directions in discourse and dialogue* (pp. 85-112). Recuperado de http://link.springer.com/chapter/10.1007/978-94-010-0019-2_5

Carberry, S., Elzer, S., & Demir, S. (2006). Information graphics: an untapped resource for digital libraries. En *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 581-588). Recuperado de http://dl.acm.org/citation.cfm?id=1148270

Carberry, S., Elzer, S., Green, N., McCoy, K., & Chester, D. (2004). Extending document summarization to information graphics. En *Proc. of the ACL-04 Workshop: Text Summarization Branches Out* (pp. 3-9). Recuperado de http://www.aclweb.org/anthology/W04-1002.pdf

Chali, Y., & Kolla, M. (2004). Summarization techniques at DUC 2004. En *Proceedings of the document understanding conference* (pp.

105-111). Recuperado de http://duc.nist.gov/pubs/2004papers/uleth.chali.pdf

Chambers, L. D. (Ed.). (1998). *Practical handbook of genetic algorithms: complex coding systems* (vol. 3). Boca Raton, FL: CRC press.

Chuang, W. T., & Yang, J. (2000). Text summarization by sentence segment extraction using machine learning algorithms. En T. Terano, H. Liu, A. L. P. Chen (Eds.), *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 454-457). Recuperado de http://link.springer.com/chapter/10.1007/3-540-45571-X_52

CICLing. Conference on Intelligent Text Processing and Computational Linguistics (2000-2014). *Recurso web de la Conferencia de procesamiento de texto inteligente y lingüística computacional*. Recuperado de http://cicling.org/

Cristea, D., Postolache, O., & Pistol, I. (2005). Summarisation through discourse structure. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 632-644). Recuperado de http://link.springer.com/chapter/10.1007/978-3-540-30586-6_70

Copernic Inc. (2013). *Página web de la herramienta Copernic*. Recuperado de http://www.copernic.com/en/products/summarizer

Corston, S., Ringger, E., Gamon, M., & Campbell, R. (2004). Task-focused summarization of email. En *ACL-04 Workshop: Text Summarization Branches Out* (pp. 43-50). Recuperado de http://www1.cs.columbia.edu/~lokesh/pdfs/Corston.pdf

Darwin, C., & De Beer, S. G. (1956). *The origin of species by means of natural selection: or, the preservation of favoured races in the struggle for life* (No. QH365. O2 1956). Recuperado de http://www.darwingame.org/origin%20annotated.pdf

D'Avanzo E., Elia A., Kuflik T., Vietri S. (2007). LAKE System at DUC-2007. En *Proc. of Document Understanding Conference 2007*. Recuperado de http://duc.nist.gov/pubs/2007papers/usalerno.pdf

Denicia, C., Montes, M., Villaseñor, L., & Hernández, R. G. (2006). A text mining approach for definition question answering. En T. Salakoski,

F. Ginter, S. Pyysalo, T. Pahikkala (Eds.), *Advances in Natural Language Processing* (pp. 76-86). Recuperado de http://link.springer.com/chapter/10.1007/11816508_10

Diao, Q., & Shan, J. (2006). A new web page summarization method. En *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 639-640). Recuperado de http://dl.acm.org/citation.cfm?id=1148294

DUC. Document understanding conference (2014). *Página web de la conferencia DUC*. Recuperado de http://www-nlpir.nist.gov/projects/duc

Eshelman, L. J. (1991). The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. En G. Rawlins (Ed.), *Foundations of Genetic Algorithms* (pp. 265–283). San Francisco, CA: Morgan Kaufmann.

Evans, D., & McKeown, K. (2005). Identifying Similarities and Differences Across Arabic and English News. Recuperado de http://hdl.handle.net/10022/AC:P:20551

Farzindar, A., & Lapalme, G. (2004). Legal text summarization by exploration of the thematic structures and argumentative roles. En *Text Summarization Branches Out Workshop held in conjunction with ACL* (pp. 27-34). Recuperado de http://www.aclweb.org/old_anthology/W/W04/W04-1006.pdf

Ferrández, S., & Ferrández, A. (2007). The negative effect of machine translation on cross–lingual question answering. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 494-505). Recuperado de https://rua.ua.es/dspace/bitstream/10045/7329/3/SFE-CICling07.pdf

Filippova, K., Mieskes, M., Nastase, V., Ponzetto, S. P., & Strube, M. (2007). Cascaded filtering for topic-driven multi-document summarization. En *Proceedings of the Document Understanding Conference* (Vol. 2007). Recuperado de http://duc.nist.gov/pubs.html#2007.

Futrelle, R. P. (2004). Handling figures in document summarization. En *Proc. of the ACL-04 Workshop: Text Summarization Branches Out*

(pp. 61-65). Recuperado de http://www.ccs.neu.edu/home/futrelle/pubs37/diagrams/DiagramPapers/futrelle-acl04.pdf

Galicia, S. N. & Gelbukh, A. (2007). *Investigaciones en análisis sintáctico para el español*. Instituto Politécnico Nacional, Dirección de Publicaciones. Recuperado de http://www.gelbukh.com/libro-investigaciones/LibroSint.pdf

García, R. A. (2007). *Desarrollo de Algoritmos para el Descubrimiento de Patrones Secuenciales Maximales* (Tesis Doctoral). Instituto Nacional de Astrofísica, Óptica y Electrónica. Puebla, México.

García, R. A., Martínez, J. F., & Carrasco, J. A. (2004). A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text. En A. Sanfeliu, *et al.* (Eds.), *Iberoamerican Congress on Pattern Recognition*, vol. 3287, (pp. 478–486). Mexico: Springer Berlin Heidelberg. Recuperado de http://link.springer.com/chapter/10.1007/978-3-540-30463-0_60

García, R. A., Martínez, J. F., & Carrasco, J. A. (2006). A new algorithm for fast discovery of maximal sequential patterns in a document collection. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 514-523). Recuperado de http://link.springer.com/chapter/10.1007/11671299_53

García, R. A., & Ledeneva, Y. (2013). Single extractive text summarization based on a genetic algorithm. En J. A. Carrasco, *et al.* (Eds.), *Mexican Conference on Pattern Recognition* (pp. 374-383). Recuperado de http://link.springer.com/chapter/10.1007/978-3-642-38989-4_38

Gelbukh, A., & Bolshakov, I. A. (2003). Internet, a true friend of translator. *International Journal of Translation*, 15(2). Recuperado de http://www.gelbukh.com/CV/Publications/2003/Bahri-2003.htm

Gelbukh, A., & Bolshakov, I. A. (2004). On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms. En, R. Monroy, G. Arroyo, L.E. Sucar y H. Sossa (Eds.), *International Atlantic Web Intelligence Conference* (pp. 105-114). Recuperado de http://link.springer.com/chapter/10.1007/978-3-540-24694-7_44

Gelbukh, A., & Sidorov, G. (2002). Automatic selection of defining vocabulary in an explanatory dictionary. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 300-303). Mexico: Springer.

Gelbukh, A., & Sidorov, G. (2006). *Procesamiento automático del español con enfoque en recursos léxicos grandes*. Recuperado de http://www.gelbukh.com/libro-procesamiento-2/

Gelbukh, A., Sidorov, G., & Han, S. Y. (2003). Evolutionary approach to natural language word sense disambiguation through global coherence optimization. *WSEAS Transactions on Computers*, 2(1). Recuperado de http://nlp.gelbukh.com/~gelbukh/CV/Publications/2003/WSD-GA.pdf

Gelbukh, A., Sidorov, G., Han, S. Y., & Hernández, E. (2004). Automatic enrichment of very large dictionary of word combinations on the basis of dependency formalism. En *Mexican International Conference on Artificial Intelligenc*e (pp. 430-437). Recuperado de http://link.springer.com/chapter/10.1007/978-3-540-24694-7_44

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Hassan, S., Mihalcea, R., & Banea, C. (2007). Random walk term weighting for improved text classification. En *International Journal of Semantic Computing*, 1(04), 421-439.

Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. New York, NY: John Wiley & Sons.

Hernández, E., García, R. A., Carrasco, J. A., & Martínez, J. F. (2006). Document clustering based on maximal frequent sequences. En T. Salakoski, *et al*. (Eds.), *Advances in Natural Language Processing* (pp. 257-267). Recuperado en http://link.springer.com/chapter/10.1007/11816508_27

Holland, J. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Recuperado en http://psycnet.apa.org/index.cfm?fa=search.displayRecord&uid=1975-26618-000

Hovy, E. (2003). Text Summarization. En R. Mitkov (Ed.), *The Oxford handbook of Computational Linguistics*. Recuperado de http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199276349.001.0001/oxfordhb-9780199276349-e-32

Kolla, M., & Chali, Y. (2005). Experiments in DUC 2005. En *Proceedings of the 2005 Document Understanding Workshop*. Recuperado en http://duc.nist.gov/pubs/2005papers/uwaterloo.kolla.pdf

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. En *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 68-73). New York, NY: ACM.

Leavitt, N. (2002). "Data Mining for the Corporate Masses", *IEEE Computer Society Press*, 35(5), 22-24.

Ledo, Y., Sidorov, G., & Gelbukh, A. (2003). Tool for computer-aided Spanish word sense disambiguation. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 277-280). Mexico: Springer Berlin Heidelberg.

Li, J., Sun, L., Kit, C., & Webster, J. (2007). A query-focused multi-document summarizer based on lexical chains. En *Proc. of Document Understanding Conference*. Recuperado de http://duc.nist.gov/pubs.html#2007.

Lin, C. Y. (2004). *Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?* Recuperado en http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings4/OPEN/NTCIR4-OPEN-LinCY.pdf

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. En *Text summarization branches out: Proceedings of the ACL-04 workshop* (Vol. 8). Recuperado de http://users.dsic.upv.es/~dpinto/duc/RougeLin.pdf

Lin, C. Y., & Hovy, E. (1997). Identifying topics by position. En *Proceedings of the fifth conference on applied natural language processing* (pp. 283-290). Stroudsburg, PA: Association for Computational Linguistics.

Lin, C. Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. En *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 71-78). Stroudsburg, PA: Association for Computational Linguistics.

Lin, C. Y., & Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. En *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Recuperado de https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/coling2004.pdf

Lin, C. Y., & Och, F. J. (2004). Orange: a method for evaluating automatic evaluation metrics for machine translation. En *Proceedings of the 20th international conference on Computational Linguistics* (p. 501). Stroudsburg, PA: Association for Computational Linguistics.

Liu, D., He, Y., Ji, D., & Yang, H. (2006). Multi-document summarization based on BE-Vector clustering. En A. Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 470-479). Recuperado en http://link.springer.com/chapter/10.1007/11671299_49

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4), 309-317.

Madnani, N., Zajic, D., Dorr, B., Ayan, N. F., & Lin, J. (2007). Multiple alternative sentence compressions for automatic text summarization. En *Proceedings of DUC*. Recuperado en http://duc.nist.gov/pubs.html#2007

Manning, C. (2007). An Introduction to Information Retrieval. Cambridge: Cambridge University Press.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing (vol. 999)*. Cambridge: MIT press.

Marcu, D. (2001). Discourse-based summarization in duc-2001. En *Proceedings of the 2001 Document Understanding Conference* (DUC-2001). Recuperado en http://duc.nist.gov/pubs.html#2001

McKeown, K., Barzilay, R., Chen, J., Elson, D. K., Evans, D. K., Klavans, J., Nenkova, A., Schiffman, B., & Sigelman, S. (2003). Recuperado de http://www.cs.columbia.edu/~delson/pubs/hlt03.pdf

Melanie, M. (1999). *An Introduction to Genetic Algorithms (eBook)*. Recuperado de https://svn-d1.mpi-inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Mihalcea, R. (2004a). Graph-based ranking algorithms for sentence extraction, applied to text summarization. En *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (p. 20). Recuperado en http://dl.acm.org/citation.cfm?id=1219064

Mihalcea, R. (2006). Random walks on text structures. En A Gelbukh (Ed.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 249-262). Recuperado en http://link.springer.com/chapter/10.1007%2F11671299_27

Mihalcea, R., & Tarau, P. (2004). *TextRank: Bringing order into texts*. Association for Computational Linguistics. Recuperado en http://digital.library.unt.edu/ark:/67531/metadc30962/

Montes, M., Gelbukh, A., & López, A. (2001). Mining the news: trends, associations, and deviations. *Computación y Sistemas*. 5 (). Recuperado en http://revistas.unam.mx/index.php/cys/article/view/2539/2101

Montes, M., Gelbukh, A., & López, A. (2002). Text mining at detail level using conceptual graphs. En U. Priss, D. Corbett & G Angelova. (Eds.), *International Conference on Conceptual Structures* (pp. 122-136). Recuperado de http://link.springer.com/chapter/10.1007/3-540-45483-7_10

Morris, J. & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, 17(1), 21-48.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Upper Saddle River, NJ: Pearson Education.

Nenkova, A. (2006). Understanding the process of multi-document summarization: content selection, rewriting and evaluation (Tesis doctoral) Columbia University, Columbia, EE.UU.

Nenkova, A., & Passonneau, R. J. (2004). *Evaluating Content Selection in Summarization: The Pyramid Method*. Recuperado en http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.1600&rep=rep1&type=pdf

Nenkova, A., Siddharthan, A., & McKeown, K. (2005). Automatically learning cognitive status for multi-document summarization of newswire. En *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 241-248). Stroudsburg, PA: Association for Computational Linguistics.

Nenkova, A., & Vanderwende, L. (2005). *The impact of frequency on summarization*. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101.

Neto, J. L., Freitas, A. A., & Kaestner, C. A. (2002). Automatic text summarization using a machine learning approach. En Bittencourt, G., Ramalho, G. (Eds.) *Brazilian Symposium on Artificial Intelligence* (pp. 205-215). Recuperado en http://link.springer.com/chapter/10.1007/3-540-36127-8_20

Open Text Summarizer (OTS) (2013). Página web de la herramienta OTS. Recuperado de http://www.splitbrain.org/services/ots.

O'Reilly, U. M., Yu, T., Riolo, R., & Worzel, B. (Eds.) (2006). *Genetic programming theory and practice II (vol. 8)*. New York, NY: Springer Science & Business Media.

Otterbacher, J., Radev, D., & Kareem, O. (2006). News to go: hierarchical text summarization for mobile devices. En *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 589-596). New York, NY: ACM.

Pertinence Summarizer (2010). *Página principal de la herramienta Pertinence Summarizer*. Recuperado de http://pertinence.net/index_en.html.

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.

Radev, D. R., Tam, D., & Erkan, G. (2003). Single-document and multi-document summary evaluation using Relative Utility. En *Proc. of the ACM International Conference on Information and Knowledge Management* (pp. 508-511). New York, NY: ACM.

Reeve, L. H., & Han, H. (2007). A term frequency distribution approach for the duc-2007 update task. En *Proc. of Document Understanding Conference*. Recuperado de http://duc.nist.gov/pubs.html#2007.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of Reading*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5), 513-523.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613-620.

Seki, Y. (2002). Sentence Extraction by tf/idf and position weighting from Newspaper Articles. *Proc. of the Third NTCIR Workshop*. Recuperado de http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-SekiY.pdf

Shrestha, L., & McKeown, K. (2004). Detection of question-answer pairs in email conversations. En *Proceedings of the 20th international conference on Computational Linguistics* (p. 889). Straudsburg, PA: ACL.

Sidorov, G., S (2015). *Grigori Sidorov CV and web-page*. Recuperado de www.cic.ipn.mx/~sidorov/

Silber, H. G., & McCoy, K. F. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4), 487-496.

Song, Y. I., Han, K. S., & Rim, H. C. (2004). A term weighting method based on lexical chain for automatic summarization. En *International*

Conference on Intelligent Text Processing and Computational Linguistics (pp. 636-639). Berlin: Springer Berlin Heidelberg.

Soricut, R., & Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. En *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 149-156). Straudsburg, PA: ACL.

Svhoong Summarizer (2013). *Página principal de Svhoong*. Recuperado de http://es.shvoong.com/summarizer

Teufel, S., & Van Halteren, H. (2004a). *Agreement in Human Factoid Annotation for Summarization Evaluation*. Recuperado de http://www.lrec-conf.org/proceedings/lrec2004/pdf/723.pdf

Teufel, S., & Van Halteren, H. (2004b). *Evaluating Information Content by Factoid Analysis: Human annotation and stability*. Recuperado de https://www.cl.cam.ac.uk/~sht25/papers/emnlp04.pdf

Tools4noobs Summarizer. (2013). Página principal de tools4noobs. Recuperado de http://www.tools4noobs.com/summarize/.

Vandeghinste, V., & Pan, Y. (2004). Sentence compression for automated subtitling: A hybrid approach. En *Proceedings of the ACL workshop on Text Summarization* (pp. 89-95). Recuperado de http://www.aclweb.org/anthology/W04-1015.pdf

Verma, R., Chen, P., & Lu, W. (2007). A semantic free-text summarization system using ontology knowledge. En *Proc. of Document Understanding Conference*. Recuperado de http://duc.nist.gov/pubs.html#2007.

Villatoro, E., Villaseñor, L., & Montes, M. (2006). Using Word Sequences for Text Summarization. En P. Sojka, I. Kopecek & K. Pala (Eds.), *International Conference on Text, Speech and Dialogue*, (pp. 293-300). Recuperado de http://link.springer.com/chapter/10.1007/11846406_37

Wan, S., & McKeown, K. (2004). Generating overview summaries of ongoing email thread discussions. En *Proceedings of the 20th inter-*

national conference on Computational Linguistics (p. 549). Recuperado de http://dl.acm.org/citation.cfm?id=1220434

Xu, W., Li, W., Wu, M., Li, W., & Yuan, C. (2006). Deriving event relevance from the ontology constructed with formal concept analysis. En A. Gelbukh, (Eds.), *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 480-489). Recuperado de http://link.springer.com/chapter/10.1007/11671299_50

Zhou, Q., Sun, L., & Nie, J. Y. (2005). IS_SUM: A multi-document summarizer based on document index graphic and lexical chains. En *Proc. of Document Understanding Conference 2005*. Recuperado de http://duc.nist.gov/pubs.html#2005

ANEXOS

Anexo A. Lista de palabras vacías

a	doesn't	is	other	there
about	done	isn't	our	theirs
after	due	it	out	them
again	during	its	over	then
all	each	it's	overall	there's
almost	either	itself	per	these
also	enough	just	perhaps	they
although	especially	kg	possible	this
always	etc.	km	previously	those
am	even	largely	quite	through
among	ever	like	rather	thus
an	first	made	really	to
and	followed	mainly	regarding	under
another	following	make	resulted	until
any	for	may	resulting	up
approximately	found	max	same	upon
are	from	me	seem	use
as	further	might	seen	used
at	give	more	several	using
be	given	most	she	various
because	giving	mostly	should	very
been	had	must	show	was
before	hardly	my	showed	we
being	has	myself	shown	were
between	have	nearly	shows	what
both	having	neither	significant	when
but	here	no	significantly	whereas
by	he	nor	since	which
can	he's	not	so	who
can't	her	now	some	while
could	his	obtain	somehow	with
couldn't	how	obtained	such	within
did	however	of	suggest	without
didn't	if	often	than	would
do	I'm	on	that	you
don't	in	only	the	
does	into	or	their	

Anexo B. Ejemplos de resultados obtenidos

Resultados detallados para los tres mejores resultados de la tabla V.3 (ver experimento 1):

Resultado primero: M, 1, *best*

1 ROUGE-1 Average_R: 0.44128 (95%-conf.int. 0.43352 - 0.44889)
1 ROUGE-1 Average_P: 0.45609 (95%-conf.int. 0.44790 - 0.46415)
1 ROUGE-1 Average_F: 0.44840 (95%-conf.int. 0.44047 - 0.45615)

1 ROUGE-2 Average_R: 0.18676 (95%-conf.int. 0.17845 - 0.19498)
1 ROUGE-2 Average_P: 0.19341 (95%-conf.int. 0.18455 - 0.20230)
1 ROUGE-2 Average_F: 0.18994 (95%-conf.int. 0.18135 - 0.19849)

1 ROUGE-SU4 Average_R: 0.20883 (95%-conf.int.-0.20138 - 0.21582)
1 ROUGE-SU4 Average_P: 0.21618 (95%-conf.int. 0.20873 - 0.22331)
1 ROUGE-SU4 Average_F: 0.21235 (95%-conf.int. 0.20483 - 0.21947)

Resultado segundo: W, f, best

1 ROUGE-1 Average_R: 0.44609 (95%-conf.int. 0.43850 - 0.45372)
1 ROUGE-1 Average_P: 0.45953 (95%-conf.int. 0.45160 - 0.46749)
1 ROUGE-1 Average_F: 0.45259 (95%-conf.int. 0.44479 - 0.46048)

1 ROUGE-2 Average_R: 0.19451 (95%-conf.int. 0.18664 - 0.20256)
1 ROUGE-2 Average_P: 0.20048 (95%-conf.int. 0.19229 - 0.20892)
1 ROUGE-2 Average_F: 0.19740 (95%-conf.int. 0.18936 - 0.20566)

1 ROUGE-SU4 Average_R: 0.21420 (95%-conf.int. 0.20755 - 0.22133)
1 ROUGE-SU4 Average_P: 0.22085 (95%-conf.int. 0.21387 - 0.22813)
1 ROUGE-SU4 Average_F: 0.21742 (95%-conf.int. 0.21061 - 0.22462)

Resultados tercero: W, f, 1*best+first*

1 ROUGE-1 Average_R: 0.46576 (95%-conf.int. 0.45877 - 0.47292)
1 ROUGE-1 Average_P: 0.48278 (95%-conf.int. 0.47547 - 0.49004)
1 ROUGE-1 Average_F: 0.47399 (95%-conf.int. 0.46693 - 0.48132)

1 ROUGE-2 Average_R: 0.21690 (95%-conf.int. 0.20915 - 0.22497)
1 ROUGE-2 Average_P: 0.22495 (95%-conf.int. 0.21659 - 0.23345)
1 ROUGE-2 Average_F: 0.22080 (95%-conf.int. 0.21278 - 0.22909)

1 ROUGE-SU4 Average_R: 0.23330 (95%-conf.int. 0.22668 - 0.24045)
1 ROUGE-SU4 Average_P: 0.24207 (95%-conf.int. 0.23508 - 0.24941)
1 ROUGE-SU4 Average_F: 0.23754 (95%-conf.int. 0.23075 - 0.24472)

Anexo C. Ejemplos de Secuencias Frecuentes Maximales

Flights were cancelled	Personal property
Sunday night	The Yucatan peninsula
Prensa Latina	Tropical storm
Civil defence	Low pressure
Hurricane Gilbert	San Francisco area
The Dominican Republic	Have to pay
The south coast	Insurance companies
The Cayman Islands	Long term
Cancun and Cozumel	State farm
In Mexico City	The big mac
Quintana Roo state	A special session
Over the water	To deal with
Year after year	Department of transportation
The white house	Gasoline tax increase
Structural damage	Might collapse in an earthquake
The California earthquake	The royal marine's music school
Bush and his aides	Military installations
The insurance industry	Private security
Exposure to catastrophes	Irish republican army
On an inflation adjusted basis	Opening day record for
Roamed the streets of Cancun	Restaurant in Moscow
Leader of the Conservative Party	The Soviet Union
The national weather service said	Moscow McDonalds
Above and beyond the usual guest	Walmart discount city
Have earthquake insurance	Vote for Major
Caused coastal flooding	Major was elected
Was elected to parliament	Associated Press
Earthquake insurance	Most important retailer of his generation
Plenty of experience	British Prime Minister John Major
For the purpose of	Vice president of the United States
Whenever I needed him	

Anexo D. Ejemplos de resúmenes generados automáticamente

En este anexo se muestran dos noticias de la colección DUC-2002, cada uno con los resúmenes realizados por humanos expertos y con el resumen generado automáticamente por el experimento 1.

Texto original A

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo. Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast. There were no reports of casualties. San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night. On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. Residents returned home, happy to find little damage from 80 mph winds and sheets of rain. Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane. The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Resumen modelo compuesto por experto 1

Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph. The Dominican Republic's Civil Defence alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Resumen modelo compuesto por experto 2

Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas. Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

Resumen generado automáticamente por el experimento 1

The national weather service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Hurricane Gilbert swept toward the Dominican Republic Sunday, and the civil defence alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. "There is no need for alarm," civil defence director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement.

Texto original B

The Irish Republican Army claimed responsibility for a huge explosion Friday that reduced a three-story military barracks on the southeast coast of England to rubble, killing 10 people and injuring 22, eight seriously. It would be one of the outlawed IRA's deadliest attacks on the main British island. Nine marine musicians and one civilian died in the blast, which also damaged dozens of nearby homes and could be heard two miles away. The musicians were between the ages of 16 and 20 as are most of the recruits in the school. A police spokesman said forensic experts are still trying to determine with certainty that the explosion was the result of a bomb. But he said the characteristics of the blast and a statement claiming responsibility appeared to confirm that it was the work of the IRA. Security sources said they believe that at least two IRA "active service" units, each composed of four or five members, are operating in Britain and continental Europe. One member, known as the "Jackal" after the assassin in the Frederick Forsyth novel "The Day of the Jackal," has been eluding the authorities for two years. He has been identified as Patrick Sheehy and has been linked to the IRA's last successful mainland bombing attack -- on an army barracks at Mill Hill in August, 1988. One soldier was killed in that incident. Sheehy and another wanted Irishman, John Conaghty, were linked to an IRA bomb factory in North London that the police stumbled upon last December while in pursuit of a car thief. A search turned up automatic and semiautomatic weapons, ammunition, 150 pounds of Semtex high explosive and a "hit list" of 100 British political figures and other officials headed by Prime Minister Margaret Thatcher. Friday's explosion occurred about 8:30 a.m. in a lounge at the Royal Marines School of Music near Deal, on the English Channel in the county of Kent. At the school are about 250 recruits who receive military and musical training before joining Royal Marines bands. The roof of the three-story barracks collapsed, trapping victims beneath the rubble. Firefighters used thermal cameras and dogs to search the debris for victims and survivors. Heavy lifting gear was brought to the scene from a nearby site where a tunnel is being built beneath the English Channel. Rescuers shouted for quiet as they used high-technology listening equipment in an effort to trace the sound of faint heartbeats. "I looked up from the sink and I just saw the whole building explode," Heather Hackett, a 26-year-old Deal housewife, told the British Press Assn. She said she told her children to run for cover, but as they did, her kitchen window shattered. "The whole window was blown across the kitchen," Hackett recalled. Her 2-year-old son, Joshua, was hit by a shard that embedded itself in his back but caused no serious injury. "I just screamed and ran out of the room," she said. "The bang was

so loud I thought the whole house was coming in." 'Appalling Outrage' Defence Secretary Tom King visited the scene and called the bombing "an appalling outrage committed against unarmed bandsmen -- people who worked for charity, who have given great enjoyment to millions right across the country, right across the world." The real evil of these murders is that the people who commit them, the 'godfathers' who send them to commit them, know that they will actually achieve nothing. Terrorism is not going to win. We shall find the people responsible for this outrage sooner or later, as we have already found some of those responsible for the earlier outrages, and they will be brought to justice." The authorities have been on high alert, expecting IRA attacks in connection with last month's 20th anniversary of the introduction of British troops into Northern Ireland. The republican underground organization opposes British rule in the predominantly Protestant province and is fighting to join the mainly Roman Catholic south in a united, independent Ireland. Visit to Ulster But in a statement telephoned to a Dublin news agency, Ireland International, Friday's attack was linked to Thatcher's visit last week to units of the controversial Ulster Defence Regiment in Northern Ireland. The locally recruited, overwhelmingly Protestant Ulster Defense Regiment has come under fire in connection with an investigation into the leak of secret government lists of suspected IRA members to Protestant assassination squads. It is widely hated by the Catholic minority in the province, and the Irish government in Dublin has urged Britain to disband the force. "Mrs. Thatcher visited Ireland with a message of war at a time when we want peace," the statement claiming responsibility for the Deal attack said. "Now in turn we have visited the Royal Marines in Kent. But we still want peace, and we want the British government to leave our country." The statement was signed "P. O'Neill, Irish Republican Publicity Bureau," a signature that has appeared on earlier IRA bombing claims. Friday's attack was the worst on the mainland since the virtually simultaneous bombings of July, 1982, directed at ceremonial military units in London's Hyde Park and Regent's Park. Eleven bandsmen and mounted guards were killed in those incidents. Eight persons were killed by IRA car bombs outside Harrods department store here in December, 1983, and 21 were killed and 162 injured in two Birmingham public house bombings in the fall of 1974. An attempted barracks bombing was averted last February when a sentry came upon two intruders who had managed to get inside a military camp in Shropshire. There has been a series of bomb and automatic rifle attacks this year on British soldiers and their families stationed in West Germany. Earlier this month an IRA gunman shot to death an army wife, Heidi Hazell, 25, in her car near her home at Dortmund.

Resumen modelo compuesto por experto 1

A huge explosion yesterday in the lounge of the Royal Marines School of Music killed ten and injured 22, eight seriously. The School is located in Deal on the English Channel. Eyewitness accounts of neighbors attest to the strength of the blast. Investigators said that it was probably a bomb blast, and the IRA has claimed responsibility. The British think that at least two IRA "active service" units, each with four or five members, operate in Britain and continental Europe. Increased IRA activity had been anticipated because last month marked the 20th anniversary of British troops entering Northern Ireland.

Resumen modelo compuesto por experto 2

In what they said was a response to Prime Minister Thatcher's "declaration of war" in a speech to the Ulster Defense Force, the Irish Republican Army claimed responsibility for an explosion which leveled a three-story barracks in Deal, killing 10 and injuring 22. The barracks, which belonged to the Royal Marines Music School, was the latest in a series of IRA bombings of military facilities. Security forces believe at least two IRA "active service units" are operating in Britain and Europe. Two members of these groups, Patrick Sheehy, known as the Jackal, and John Conaghty are being sought in connection with earlier attacks.

Resumen generado automáticamente por el experimento 1

(Was obtained automatically using the proposed method from Experiment 1). The Irish republican army claimed responsibility for a huge explosion Friday that reduced a three-story military barracks on the southeast coast of England to rubble, killing 10 people and injuring 22, eight seriously. It would be one of the outlawed IRA's deadliest attacks on the main British island. Nine marine musicians and one civilian died in the blast, which also damaged dozens of nearby homes and could be heard two miles away. The musicians were between the ages of 16 and 20 as are most of the recruits in the school. A police spokesman said forensic experts are still trying to determine with certainty that the explosion was the result of a bomb.

Generación automática de resúmenes
Retos, propuestas y experimentos

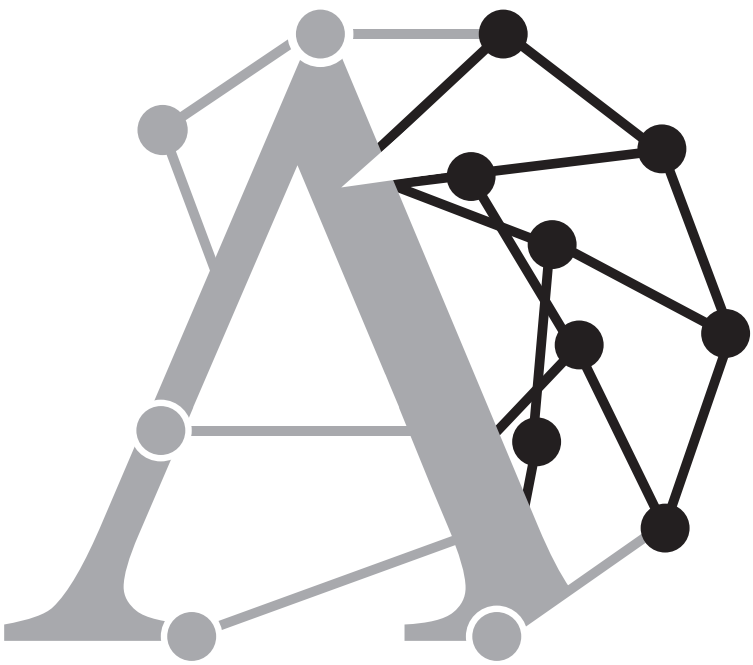
Automatic Generation of Text Summaries
Challenges, proposals and experiments

de Yulia Nikolaevna Ledeneva y René Arnulfo García Hernández, fue impreso en los talleres de Editorial CIGOME, S.A. de C.V., Vialidad Alfredo del Mazo núm. 1524, ex. Hacienda La Magdalena C.P. 50010, Toluca, México. Su edición consta de 1 000 ejemplares. La edición estuvo a cargo de la Dirección de Difusión y Promoción de la Investigación y los Estudios Avanzados.

Coordinación editorial: Patricia Vega Villavicencio
Corrección de estilo (versión en español): Tomás Fuentes Estrada
Diseño de portada e interiores: Juan Manuel García Guerrero y Cristina Mireles Arriaga

Automatic Generation of Text Summaries

Challenges, proposals and experiments



UAEM | Universidad Autónoma del Estado de México

Dr. en D. Jorge Olvera García
Rector

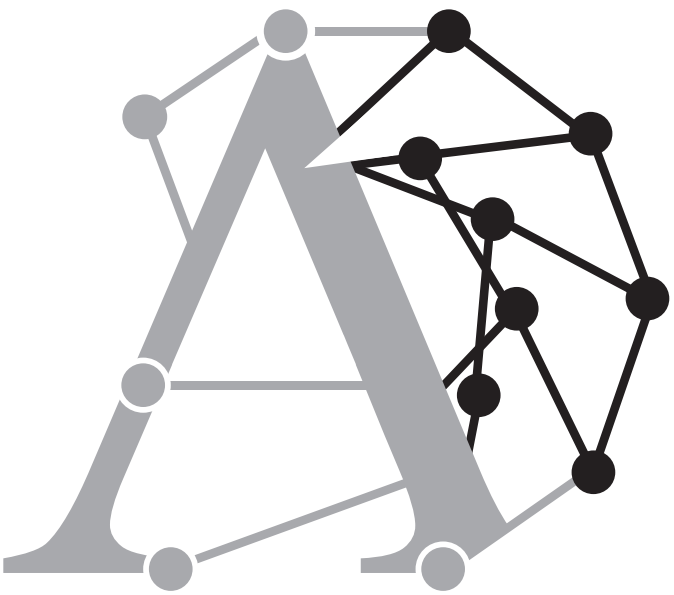
Dra. en Est. Lat. Ángeles Ma. del Rosario Pérez Bernal
Secretaria de Investigación y Estudios Avanzados

M. en A. P. Guadalupe Ofelia Santamaría González
Coordinadora de la Unidad Académica Profesional Tianguistenco

L. C. C. María del Socorro Castañeda Díaz
Directora de Difusión y Promoción de la Investigación y los Estudios Avanzados

Automatic Generation of Text Summaries
Challenges, proposals and experiments

Yulia Nikolaevna Ledeneva
René Arnulfo García Hernández



UAEM | Universidad Autónoma del Estado de México

Generación automática de resúmenes
Retos, propuestas y experimentos

Automatic Generation of Text Summaries
Challenges, proposals and experiments

Traducción del español a inglés: Luis Cejudo Espinosa

Libro de investigación arbitrado por pares ciegos, con base en los criterios establecidos por la Secretaría de Investigación y Estudios Avanzados.

1a edición, enero 2017

ISBN: 978-607-422-782-6

Instituto Literario núm. 100 Ote., Centro, C.P. 50000, Toluca, México
http://www.uaemex.mx

Impreso y hecho en México
Printed and made in Mexico

To my daughters
Renata Yulie
María Constanza

Content

List of tables

List of figures

ABSTRACT

Over the last two decades the exponential increase of electronic information has created the need to understand the essential content of large volumes of textual information. The first step is to discern what, out of such electronic information at our disposal, is of our primordial interest. In fact, it is estimated that eighty percent of the electronic information of a company is textual, while the other twenty percent exists as databases. This percentage increases when internet is involved, since most of the information there is textual, this is to say, in natural language.

This points at the importance of developing automatic methods that allow detecting the most relevant content of a document in views of producing a shorter text that works as a summary. The Automatic Generation of Text Summaries (AGTS) is one of the priority tasks in the field of research on Natural Language Processing, which intends to produce summaries as similar as possible to those written by humans. In order to do so, a number of models and methods have been proposed; from our point of view, these are impractical, as they cannot work independently from the language or domain of the text document; or else, they have to be fed with a series of resources and linguistic processes (dictionaries, taxonomies, grammars, syntactic analyzers, etc.) that require heavy intervention of human work. Nevertheless, these proposals are still interesting or promising in the future. With the intention of presenting a more complete state of the art of this research area, not only do we present a revision of the methods developed for these purposes, but also include an assessment of commercial AGTS tools.

This book presents a computing method novel at international level, which is the product of research undertaken to look for computing methods and models the least dependent on language and domain that can be applied for the Automatic Generation of Text Summaries, but surpassing the quality of those that can be produced at present.

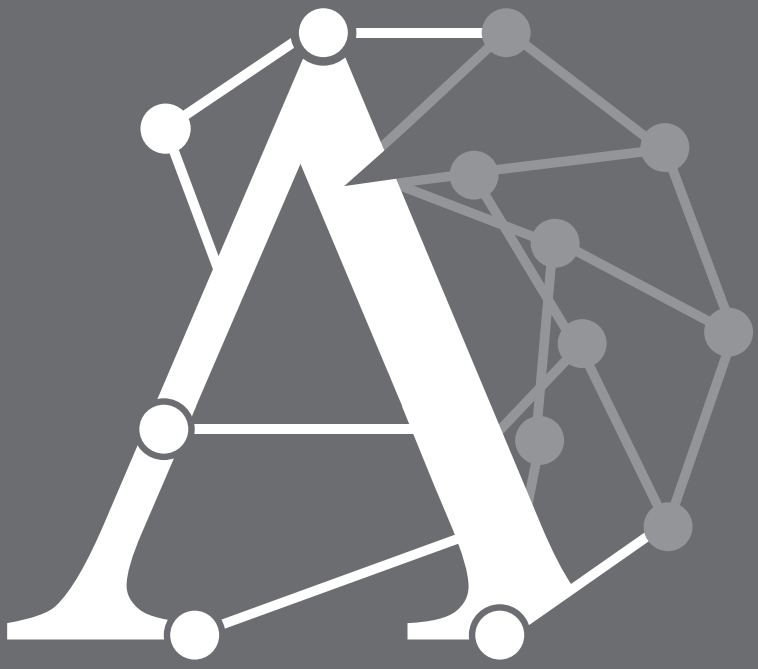
One of the contributions derived from the research was the design of a methodology that allows learning which will be the maximal quality that an AGTS model can obtain from a determinate document collection. With this parameter, so far unknown, it is possible to learn how significant the advancement of new AGTS methods and commercial tools is; and with this, it can be acknowledged how much there is still left to discover in this field of research.

Another contribution presented in this book is that we noticed that all AGTS methods present four stages in their processes: *term selection, term weighting, sentence weighting* and *sentence selection* for the summary. This allows analyzing methods previously proposed, but additionally new and better researches can be generated if innovations for each of these stages are proposed. Let us remember the axiomatic principle to solve problems, "divide and conquer". This way, in this book we present several innovations in each of the AGTS stages and the corresponding experimentations that the proposed method validates.

In short, for the first stage, *term selection*, we describe how to detect multiword descriptions that convey an important meaning. To do so, we propose to utilize the word sequences that frequently appear in the text, attempting that at a higher frequency it becomes possible to characterize what the text is about. However, as there can be many Frequent Sequences (FSs), we propose to only use Maximal Frequent Sequences (MFSs), which are no subsequence of any other frequent sequence. By using MFSs we try to enhance the meaning of each term, because one MFS represents a whole set of frequent sequences. In fact, from the set of MFSs it is possible to obtain all the frequent sequences.

Also for the stages of *term weighting, sentence weighting* and *sentence selection* there are innovations that enable us to develop a computing method for the AGTS problem. The new method developed in this book has obtained better results than the state of the art, considering collections of standard items of news that are used to try the new proposed methods at international level.

Students and researchers in the fields of natural language processing, artificial intelligence, computational science and computational linguistics may be the first to take an interest in this book. However, we also intend to introduce the book to the general public in this promising research area; this way, we have taken the liberty to translate to Spanish, in everyday language, some technical words and Anglicisms proper to this research area, but mentioning, at all times, the English language terms for those readers interested in broadening their sources of knowledge.



CHAPTER I.
BASIC CONSIDERATIONS

In this chapter we present some basic concepts necessary to clearly present the research problem approached in this book. Then, we briefly expose the objectives pursued in the development of the new method and the methodology that enables us to reach such objectives; those that can be learnt in this book are described as well. Finally, the organization of this work is included.

I.1 Introduction

Over the last two decades the exponential increase of electronic information has made it necessary to understand the essential content of large volumes of textual information. The first step is to discern what out of this electronic information at our disposition is of our primordial interest. In fact, it is estimated that eighty percent of the electronic information of a company is in text format and the other twenty percent exists as databases (Leavitt, 2002). This percentage pales in comparison with the Internet, as most of the information there is in textual form, this is to say, natural language.

The summary of a written text is much shorter than the original, however it conveys the most relevant or important information contained in the original. There is a series of scenarios in which the automatic construction of such summaries turns out to be useful; for instance, an information retrieval system might present the automatically generated summaries of the list of retrieved documents so that the user can quickly decide which documents interest them and are worth opening to take a closer look. To a certain extent this is what Google models with the fragments shown in its search results. Figure I.1 shows the titles or the text fragments retrieved by Google from the original texts in which the words "summary generation" (*generación de resumen*) appear. Of the first four documents retrieved, only the first of them would be interesting for the present work. However, the other 34,300,000 documents in which the information we seek may be found would have to be revised.

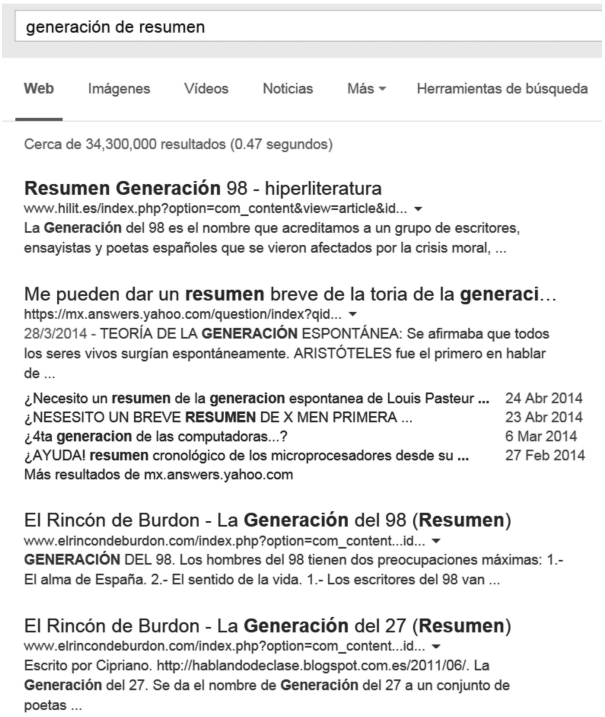


Figure I.1 First four documents retrieved from Google with the query "*generación de resumen*" on May 8th, 2014.

Other instances include the automatic construction of journalistic articles, of which the very news agencies can handle hundreds of news items on various categories (Evans, McKeown, 2005), (McKeown, 2003), (Nenkova, Siddharthan, McKeown, 2005). For instance, Notimex handles more than 500 news items a day; also, companies and organizations receive several email messages, which can be summarized to be sent to mobile devices as a text message (SMS) (Corston-Oliver, Ringger, Gamon, Campbell, 2004), (Shrestha, McKeown, 2004), (Wan, McKeown, 2004). Nowadays, political parties, presidential candidates and companies specialized in launching new products, such as films or songs, have the need to learn instantly as a summary what the social media are saying about them. However, in social media with heavy traffic, namely Twitter and Facebook, under certain circumstances there can be thousands of messages published a second, this makes it impossible for a human being to read all of them in short time. Because of this, it is necessary to have summary-generating tools which, together with opinion mining tools, can meet such demands.

This need to summarize information is shared by researchers in various fields of knowledge in which new technological advancements are published in journals and specialized congresses, and also in activities in which there is a great deal of available information, as in legal texts (Farzindar, Lapalme, 2004). These days, personal mobile devices such as smartphones and tablets allow having large amounts of textual information available, either previously downloaded or consulted at the moment on the Internet. However, the screens of these devices can be very small, so reading a full document can be tiresome. An AGTS tool that works online or that can be installed in the mobile device can help solve this problem (Futrelle, 2004), (Dia, Shan, 2006), (Otterbacher, Radev, Kareem, 2006), (Carberry, et al., 2004), (Carberry, Elzer, Demir, 2006).

As it is noticed, the needs of AGTS are diverse in relation to their contexts, devices and users. This is why in this research the development of generic methods to summarize is considered indispensable; these methods shall depend the least on the domain and language of the input text, but only do on those in which the length of the desired summary can be controlled.

Nowadays, there are several AGTS tools both freely and commercially available that allow generating summaries; among them one finds: *Copernic Summarizer* (Copernic, 2013), *Microsoft Office, Svhoong Summarizer* (Svhoong, 2013), *Pertinence Summarizer* (Pertinence, 2010), *Tools4noobs Summarizer* (Tool4noobs, 2013) and *Open Text Summarizer* (Ots, 2013). Some of these can only work in determinate languages, nevertheless. Part of this book will also focus on learning the characteristics of these tools and assess the quality of the generated summaries. With this, it will be possible to learn what the advancement of AGTS tools is in comparison with methods reported as state of the art.

AGTS is a research task of the Natural Language Processing (NLP) area, which is part of a multidisciplinary field in which linguistics, mathematics, statistics, computing, pattern recognition, data mining, artificial intelligence, among others, concur. All these disciplines make it possible that computers understand written natural language. Were this last possible, computers would be able to understand isolated words, sounds and phrases and also grasp the ideas language has; this might help humans to carry out various tasks related to natural language from those very simple to the very complex. NLP tasks are well-differenced when they deal with spoken and written natural language; this field has been called automatic text treatment.

In automatic text treatment there can be applications from the simplest to the most complex. For instance, separate a word, detect and correct spelling and grammar mistakes, fact and coherence checkers, visua-

lization and exploration of large collections of documents, information retrieval systems, question answering, automatic translation, plagiarism detection, authorship recognition, summary generation, among others.

The methods to generate summaries can be classified according to their sort of input, in single document or in multiple documents. In Single Automatic Text Summarization there is only one summary of a single text; whereas in Multidocument Automatic Text Summarization, the summary is produced from a series of input documents (as well as the news items of the day, or the results of a query). In this book we present and experiment AGTS with a single input document.

AGTS methods can be classified according to the sort of output summary as abstractive and extractive summaries (Lin, Hovy, 1997). An abstractive summary is an arbitrary text that describes the context of the original document. The process of generating abstractive summaries consists in "understanding" the original text and "rewriting" it with fewer words. Specifically, an abstractive summary method uses a larger amount of methods and more sophisticated linguistic resources to examine and interpret the text searching for new concepts and expressions that allow describing the text in a shorter way, but preserving the most important information of the original document. Even if this seems to be the best way to produce a summary (and this is how human beings do so), in real life the current linguistic technology related to the analysis and generation of text is not mature yet for the methods to be viable in practice, considering the various domains and languages of the input text.

This way, humans use their background knowledge to generate a summary stressing that which they consider important; this makes each human being generate summaries with different words, but preserving almost always the same information. This also occurs with abstractive AGTS methods, which can produce wrong interpretations, as they depend on the quality of the information sources from which new concepts and expressions are taken, thus altering the meaning of original text.

Conversely, an extractive summary basically consists in selecting important sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order –this is to say, a copy of the original text with most sentences omitted. A method to produce extractive summaries only decides, for each sentence, whether it will be included in the summary. The resulting summary is more a list or compendium of important ideas, so the reading might not be fluid. However, the simplicity of the underlying statistical techniques makes extractive summaries an interesting and robust alternative, which might be obtained by means of the most "intelligent" extractive methods regardless of the language.

Due to the potential AGTS methods have in the humongous textual information contained in electronic media, in this book we deal with the issue of extractive GART from a single document; using the least resources and sophisticated linguistic methods, this way looking for independence from the language and domain of the input text. This does not imply that the ideas presented cannot be later adapted to work with abstractive summaries or with multiple-text inputs.

To exemplify the problem for the reader, we make the following exercise that can indeed be very interesting to perform:

Considering the following five sentences[1] from a document, which would be the two sentences chosen as a summary?

A. Egypt's government protects the pyramids
B. Egypt's pyramids are cultural heritage
C. The pyramids were built by the Pharaohs
D. Egypt's pyramids were tombs for the Pharaohs
E. A good government protects its cultural heritage

In advance, any selection made is the right choice, but as we will see further, the criterion among humans on this small collection also differs a little. Another of the fundamental aspects to consider is that you are using your background knowledge of the Spanish language to make an interpretation of it. Specifically, as for the Spanish language it was necessary to resort to lexical knowledge, the meaning of words; syntactic knowledge, the grammatical structure; and semantic knowledge, the interpretation of a message as a whole. However, the differences in culture and experiences of an individual make each stage have its own interpretations, thus generating ambiguity.

In this case, the method we propose suggests, without background or language knowledge, that you probably chose sentences D and B, in this order of importance; then, it may be C and E; while the least likely is A. This example will be developed over the book to show how we decided to propose a method with these characteristics.

A typical method to generate extractive summaries consists of a number of steps, in each of them there are various options that can be chosen. Let us suppose that the selection units are sentences (they can be, however, phrases or paragraphs). This way, the final goal of the process to generate extractive processes is sentence selection. One of the ways to select the right sentences is to assign a numerical measure of the usefulness of a sentence for the summary in order to choose the best. The process to assign the usefulness weights is called weighting; one of

[1] Originally in Spanish. TN.

the ways to estimate the usefulness of a sentence is adding the weights of each of the individual terms comprised in the sentence. The process to estimate the individual terms is called term weighting; to do so, one shall decide which the individual terms that correspond to the selection term task will be. The different methods to generate extractive summaries can be characterized according to how they perform these tasks.

In this book we present a novel way to select terms, to weight the terms and sentences and select sentences. Moreover, several simple options are analyzed for the statistical selection of terms independent from language and their corresponding term weighting, based on units larger than a single word. Particularly, in this work new terms are tried; these are called multiword descriptions, which are based on maximal frequent sequences that promise to be a good selection for the task of automatic generation of text summaries.

The originality of this work is to present a new computational method, based on the very structure of the text for an AGTS problem with little dependence on language. The tested method, which is presented in this book, has obtained results superior to those that can be obtained from current commercial tools and state-of-the-art methods. This was attained using collections of standard news items specifically created to try the new methods proposed at international level.

I.2 Book objectives

1. Identify the best methods and commercial tools developed for AGTS.
2. Identify the basic stages that all AGTS methods follow.
3. Identify the linguistic and methodological resources that are used in state-of-the-art researches on text models and AGTS methods.
4. Identify the corpuses of documents and assessment systems habitually employed to try new AGTS proposed methods.
5. Learn the quality of summaries generated by commercial AGTS tools, so they will be assessed with collections of standard news items.
6. Learn the highest possible quality that an AGTS method can attain for the assessment collection to be used.
7. Seek text models that allow selecting terms with semantically enhanced meanings, based on the discovery of multiword descriptions.
8. Develop AGTS methods with better quality than those of the state of the art. With this we expect the system to be useful for the users and to work as a framework as well.

9. Intend that the methods to be developed for AGTS depend the least on language and domain.

I.3 What we can learn from this book

- Identification of the general steps followed by a method of automatic generation of extractive text summaries.
- An introduction to the area of natural language processing from the AGTS research task.
- The associations, institutions, laboratories, congresses and researchers in Mexico that are working in the NLP area.
- Description of new methods to generate text summaries based on the extraction of new multiword terms.
- Development of new methods to generate text summaries from a single document, regardless of language and domain.
- New methods to automatically generate text summaries with results superior to those of the state-of-the-art methods.
- Identify the linguistic and methodological resources that are utilized in the state-of-the-art researches in the text models and AGTS methods.
- Identify the corpuses of documents and the assessment systems commonly utilized to try new AGTS methods.

I.4 Book organization

This book is organized in six chapters. In this first one, we introduced the research problem, the stated objectives and what is expected from the book. The following chapter summarizes the state-of-the-art methods. In the third chapter, we describe the theoretical framework. In the fourth chapter, we introduce the proposal of a method for the automatic generation of text summaries from a single document. In the fifth chapter, the experimental results are presented. And in chapter 6, one finds the conclusions.

CHAPTER II.
STATE-OF-THE-ART METHODS

This chapter offers a detailed presentation of the state of the art. However, in section II.1 we will firstly present the area of Natural Language Processing and its applications. Then, in subsection II.1.1 the state of NLP in Mexico is commented; identifying associations, laboratories, institutes, researchers and main congresses. Section II.2 presents an introduction to the state of the art in summary generation; specifically, in section II.2.1 a detailed description of the state of the art in extractive summary generation will be exposed. This section is ordered according to the four steps followed by a method of automatic generation of extractive summaries. In section II.3 we show the assessment measures that compare the quality of automatically generated summaries with human-generated summaries. Finally, in section II.4 we show the assessment of 7 commercial tools and 3 AGTS state-of-the-art methods for DUC-2002 document collection.

II.1 Natural language processing

Natural language processing (NLP) has its origins in artificial intelligence, when it was intended to provide computers with intelligence so that they were able to process natural language in views of understanding, reproducing, inferring and deducing the knowledge present in the information. At the beginning, the first models came from mathematics, statistics and linguistics, principally. Hence, the appearance of Computational Linguistics (CL), which describes the functioning models of NLP systems; it explains how to compile data for the needs of NLP systems, develops software to correct n-grams, resolves the disambiguation of the meaning of words, builds dictionaries and databases, retrieves information, translates from a language to another, etc. (Bolshakov, Gelbukh, 2004a). As many other areas (for instance, the areas of mechanics and chemistry), LC has the need of intelligent processing of tools and the automation of NLP tasks.

NLP tasks are well differenced when natural spoken language and natural written language problems are dealt with. The latter is called automatic text treatment.

In automatic text treatment one can work with applications, from the simplest to the most complex. For instance, breaking down words, detecting and correcting spelling or grammar mistakes, checking facts and coherence, visualization and exploration of large document collections, information retrieval systems, information extraction systems, question answering, automa-

tic translation, plagiarism detection, recognition of the authorship of a text, summary generation, among others.

We now describe some tasks of automatic text treatment:

- Word Sense Disambiguation (WSD) (Gelbukh, Sidorov, 2002), (Manning, Schütze, 1999): it decides which meaning or sense has any given word, generally in function of their context. This task is very important since on the success of its resolution will depend the accuracy of other tasks such as automatic translation, question answering, etc.
- Information retrieval (IR) (Manning, 2007), (Baeza-Yates, Ribeiro-Neto, 1999): it consists in the search for documents of unstructured nature that meet the need for information considering large collections of documents, generally in local systems or the Internet. This area surpasses the search in traditional databases, thus becoming the prevailing way to access information. Now hundreds of million people use IR systems everyday when they use a search engine or search in their email inboxes, for instance Google.
- Automatic translation (Gelbukh, Bolshakov, 2003), (Bolshakov, et al., 2004): it is a system that aided by computers is responsible for the translation from one language to another. This application is very useful for business and scientific purposes, because international collaboration grows exponentially.
- Question answering (QA) (Aceves-Perez, 2007), (Ferrández, Ferrández, 2007): it is a complex task, which combines the techniques of NLP and IR and automatic learning. QA main objective is to locate the right answer for a question written in natural language in an unstructured collection of documents. QA systems are similar to search engines, in which the input to the system would be a question in natural language and the output would be the answer to it (not a list of full documents as in IR).
- Document clustering (Hernández-Reyes, 2006): from a document collection on any subject, the objective is to automatically group them considering the various topics of each document in such manner that the documents in a group are very similar, but at the same time they are utterly different from documents in other groups. This sort of algorithms are very useful in other applications such as AGTS, in which similar sentences or similar document groups are found in the first place and then summarized. In IR, the documents retrieved in the query are shown by groups, which allows going directly to the users' group of interest.

II.1.1 Natural language processing in Mexico

More than 50 years have passed since the first advancements in NLP were published. Ever since, a large number of works have been carried out all over the world, including Mexico. We especially underscore the work in the Laboratory of Natural Language of the Center of Research in Computing of the National Polytechnic Institute, where more than 400 works have been produced over the last 15 years, mainly by Alexander Gelbukh (Gelbukh, 2014), Grigori Sidorov (Sidorov, 2014) and Igor Bolshakov (Bolshakov, Gelbukh, 2000), (Bolshakov, et al., 2004a). Their works establish the basic definitions and new discoveries of research in various NLP areas:

- Lexical resources (Gelbukh, Sidorov, 2006)
- Construction and compilation of dictionaries (Gelbukh, Sidorov, 2002), (Gelbukh, Bolshakov, 2003), (Gelbukh, et al., 2004a)
- Collocation database (CrossLexica) (Bolshakov, 2000), (Bolshakov, 2004b), (Bolshakov, Bolshakova, Kotlyarov, Gelbukh, 2008)
- Syntactic analysis of the Spanish language (Galicia-Haro, Gelbukh, 2007)
- Semantic errors and malapropism (Gelbukh, et al., 2004b), (Bolshakov, Galicia-Haro, Gelbukh, 2005)
- Word sense disambiguation (Gelbukh, Sidorov, Han, 2003), (Ledo, Sidorov, Gelbukh, 2003)
- Automatic translation (Gelbukh, Bolshakov, 2003)
- Text mining (Montes-y-Gómez, Gelbukh, López-López, 2001), (Montes-y-Gómez, Gelbukh, López-López, 2002)

It is fair to point out that one of the main CL conferences at international level is organized by Professor Alexander Gelbukh (CICLing, 2014). Nevertheless, other research centers have built new NLP laboratories such as the National Institute of Astrophysics, Optics and Electronics, where mainly the researchers (Montes-y-Gómez, et al., 2002), Luis Villaseñor (Denicia-Carral, et al., 2006), José Francisco Martínez Trinidad (Hernández-Reyes, et al., 2006) and Jesús Ariel Carrasco Ochoa (Hernández-Reyes, et al., 2006) have produced related works in this field. Another of the important NLP laboratories is led by researcher Gerardo Sierra Martínez in the National Autonomous University of Mexico. Other universities such as the Meritorious Autonomous University of Puebla are working as well on NLP, headed by researcher David Pinto. The Metropolitan Autonomous University with researcher Héctor Jiménez has also generated works related to this area. As a matter of fact, the interest and collaboration of these researchers has led to the foundation of the Mexican Association for Natural Language Processing (*Asociación Mexicana para el Procesamiento de Lenguaje Natural*, AMPLN) (AMPLN, 2014), of which the authors of the present work are members.

AMPLN is a nonprofit professional organization with the following objectives (AMPLN, 2014):

- Promote the interaction and interchange of ideas, tools and resources between Mexican NLP specialists.
- Represent the Mexican community of NLP specialists.
- Disseminate the achievements and importance of NLP among the national society.

The mission of AMPLN is to foster the interaction and interchange of ideas between Mexican specialists in NLP and also to disseminate the achievements and importance of NLP among the national society.

II.2 Generation of text summaries

The early experimentation by the end of the 1950's and the 1960's suggested that the generation of text summaries by a computer was viable, however not easy. After some decades, advancement in NLP, together with the growing presence of online text –in corpuses and especially in the web– have renewed the interest in the automatic generation of text summaries. This way, the large amount of electronic documents available on the Internet has motivated the development of very good information retrieval systems. However, the information provided by these systems, such as Google, only shows the part of the text in which the words of the query appear.

According to the classic viewpoint (see below how we introduce our viewpoint), there are three stages in the automatic generation of text summaries (Hovy, 2003). The first stage is carried out by means of identifying the topics; at this stage almost all the systems use a number of independent modules. Each module assigns a score to each input unit (word, sentence or longer passage); then, another module combines the scores of each unit in order to assign a single score. Finally, the system returns the highest-scored units, according to the summary length asked by the user. The performance of the modules to identify the topics is usually measured by means of the scores of recall and precision (see next section).

The second stage is denoted by (Hovy, 2003) as the interpretation stage. This stage distinguishes the extractive generation systems from those of abstractive nature. In the interpretation, the topics identified as important are merged, represented as new terms and expressed using a new formulation utilizing concepts or words that are not in the original text. No system can perform interpretation without previous knowledge of the domain. By definition, the system has to interpret the input in terms of something alien to the text. However, the previous acquisition of suffi-

ciently background knowledge of the domain is so difficult at present that AGTS systems only deal with a small part. Therefore, the disadvantage of this stage is still hindered by the problem of acquiring knowledge on the domain.

The generation of the summary is the third stage of a system that generates text summaries. Once the summary content has been created in an internal notation, techniques to generate natural language are needed, namely: text planning, sentence planning and sentence production.

We identify four stages for a computational system to generate a text summary:

Term selection: in this stage one shall decide which units will be terms, for instance, they can be words, n-grams or sentences.

Term weighting: it is a weighting (or estimation) process of the individual terms in relation to the document content.

Sentence weighting: this is the process in which a numerical measure is given to the usefulness of the sentence. For instance, one of the ways to estimate the usefulness of a sentence is to add the usefulness weights of the individual terms the sentence comprises.

Sentence selection: sentences (or other units selected as final parts of the summary) are selected. For instance, one of the simplest ways to select sentences is to assign some numerical measure to the usefulness of each sentence of the original text to later select the best for the summary.

II.2.1 Automatic generation of extractive summaries

Without considering the pre-processing of the text normally followed by the application of NLP, we present the state of the art following the four stages of an AGTS method, which were previously presented. The pre-processing used in this research is presented in the following chapter.

II.2.1.1. Term selection

According to the goals stated in the research, the most adequate terms are those least dependent on language or domain. Some of these models are the models known as bags of words (Salton, Buckley, 1988), (Salton, 1989), n-gram (Villatoro-Tello, Villaseñor-Pineda, Montes-y-Gómez, 2006) and (Ahonen-Myka, 1999), (Ahonen-Myka, 1999a), (Ahonen-Myka, 1999b), (Ahonen-Myka, 2002). However, there are also some other interesting models with heavier dependence such as, elementary discour-

se units (Marcu, 2001), (Soricut, Marcu, 2003), factoids (Teufel, Halteren, 2004a), (Teufel, Halteren, 2004b), information nuggets (Liu, He, Ji, Yang, 2006), semantic content units (Nenkova, 2006), lexical chains (Morris, Hirst, 1991), keywords or keyphrases (D'Avanzo, Elia, Kuflik, Vietri, 2007). In this case, we consider that it is adequate to present the terms that depend the most on language first, in views of understanding what sort of information such terms try to represent. Later on, after presenting the least language-dependent terms, the strengths and weaknesses of these models can be compared.

Elementary Discourse Units (EDU) were used in AGTS in Marcu's work (Marcu, 2001), (Carlson, Marcu, Okurowski, 2003). EDU are phrases or clauses present in sentences, which are determined using grammatical, lexical and syntactical information of the language that is being analyzed. For instance, there are prepositional, substantive, verbal clauses, etc., which allow recognizing parts, and in the best of cases, complete sentences. Specifically, in Marcu's (Marcu, 2001), (Carlson, et al., 2003) work substantive clauses were used.

Factoids (Teufel, Halteren, 2004a), (Teufel, Halteren, 2004b) are semantical units that represent the meaning of a sentence by means of the facts that can be deduced when these facts also appear in other documents. For example, from the sentence "The police has arrested a white Dutch Man" the following factoids can be derived: "A suspect was detained"; "The police made the arrest"; "The suspect is white"; "The suspect is Dutch"; and, "The suspect is a man". Factoids are empirically defined based on data from the set of summaries (generally, some summaries are manually made taken from the corpus (Duc, 2014)). The definition of factoid starts with the comparison of the information contained in two summaries and the factoids would add or divide in an incremental manner according to the considered summaries. If two items of information occur together in all the summaries and in the same sentence, they are considered a factoid, as the differentiation in more than a factoid will not help us to distinguish the summaries. Factoids are labeled with natural language descriptions; at first, these are more like the wording in which the factoid occurs in the first summaries, even though the annotator tries to identify and treat the paraphrase of information of the factoid alike when these occur in other summaries. If (together with a number of sentences in other summaries) a summary contains "was murdered" and another "was shot dead", the following factoids will be identified: "There was an attack", "The victim died", "A pistol was used". As it is noticed, it is necessary to carry out a deep semantic analysis of a large amount of documents in order to obtain factoids related to the document so that they are useful.

Another possible proposal, presented by Nenkova in (Nenkova, 2006) is Semantic Content Units (SCU). The definition of content unit is somewhat dynamical, it can be a single word, but never longer than a clause of a sentence. The most important evidence of their presence in a text is the information expressed in two or more summaries, i.e., the frequency of the content unit in a text. Another evidence is that these content units can frequently have a different wording (but the same semantical meaning) which creates difficulties for their extraction independently from language.

Lexical chains were introduced for the first time by (Morris, Hirst, 1991); these chains exploit the cohesion among an arbitrary number of related words. Lexical chains are formed by means of chaining or relating one of the semantical classes the words have (i.e., they have a flow in their meaning) (Barzilay, Elhadad, 1999), (Silber, McCoy, 2002). For instance, as a semantic class one can resort to identities, synonyms or the hypernym/hyponym pairs which will allow grouping them in the same lexical chain. There is a number of difficulties to determine which lexical chain or determinate word should be joined. For instance, a particular substantive instance can correspond to a various different meanings of the word and so the system shall determine which meaning has to be utilized (for example, if a particular case of "house" can be interpreted as 1: place to live; 2) legislative house). Lexical chains have been utilized in AGTS works in (Brunn, Chali, Pinchak, 2001), (Zhou, Sun, Nie, 2005), (Li, et al. 2007).

Keyphrases, also known as keywords, are linguistic units, usually, longer than a single word, but shorter than a full sentence. There are several sorts of keyphrases that go from statistical keywords (only sequences of words) to those of linguistic nature (defined in function of grammar) (D'Avanzo, et al., 2007).

As it is observed, most of the aforementioned terms consider more than a single word in their definition, trying to semantically enhance the extracted terms. In like manner, it is noticed that the various definitions try to limit or characterize how these terms can appear in the text, which, as explained, need linguistic resources that depend on language and domain.

Contrasting with the previous terms, one of the first language independent models, and easy to extract, is the utilization of single words as terms of the document; this was proposed by Salton (Salton, Wong, Yang, 1975), (Salton, Buckley, 1988), (Salton, 1989) and it is known as bag of words. Nevertheless, by considering a single word polysemy increases owing to the loss of context of the very term. Another problem that this model conveys is that there are too many terms, even for a small text.

In the face of the loss of context that bag of words implies, word sequences of a determinate n size, which are known as n-grams, have been utilized. Although the problem of high dimensionality persists, n-grams have been widely utilized in diverse AGTS researches because of their easy extraction and because they decrease the loss of context (enhancing the extracted terms at longer n sizes), making them robust for AGTS (Villatoro-Tello, et al., 2006).

In views of increasing independence from domain and language with the generated summaries, (Villatoro-Tello, et al., 2006) eliminates all sort of attributes that depend on language and domain using only features based on words. In particular, only word sequences (n-grams) are used as terms.

Even if with the first attempt to use n-grams the results of other methods are exceeded, there are some disadvantages. One of them is they are always sentences fixed in size, which was previously defined by the user. An important part of the problem in such techniques is in the definition of the size of the sequence to be extracted, which by and large depends on the analysis of the text.

Another of the interesting terms is that proposed by (Ahonen-Myka, 1999), which states to utilize sequences of words that are frequent, so this model needs a frequency threshold. Since the frequent sequences that can be extracted are a great many, only those frequent sequences that are no subsequences of any other are taken, this is to say, they are maximal. If we speak in terms of n-grams, it is the same as saying that one finds the set of all the frequent n-grams, where n goes from 1 to x (where x is the size of the longest and most frequent gram) and from there the set reduces to only the maximal. However, the sequence term goes beyond a gram, because it is not necessary that the elements appear contiguously as it is the case of gram, but it is necessary that they appear in the same sequence or order. To restrict the separation that may exist between elements that are part of a frequent sequence, a parameter known as GAP is utilized. Maximal frequent sentences (MFSs) with GAP were utilized in applications which reveal the topics of the documents (Ahonen-Myka, 1999a), (Ahonen-Myka, 1999b), (Ahonen-Myka, 2002), which can be used for information retrieval tasks. This way, MFSs can be seen as keyphrases. Although they seem to have a number of advantages on n-grams, one of their main disadvantages is the computational complexity that the extraction of MFSs brings along, as it increases according to the size of GAP (García, et al., 2004), (García, et al., 2006), (García, 2007); when GAP tends to be longer, the complexity tends to be exponential (this tells us it is one of the most complex problems computationally speaking). Even if no algorithm (already developed) can minimize its complexity, in García's work

(García, et al., 2004), (García, et al., 2006), (García, 2007) some algorithms that allow managing more efficiently the computational resources were developed in order to extract MFSs in a faster way. Since MFSs seem to be good candidates to be used as semantically enhanced terms and had not been tried for AGTS, in this research our hypothesis is that they can improve the automatically generated summaries.

II.2.1.2 Term weighting

Out of the works devoted to methods based on terms, the majority concentrates on term weighting. The terms identified in the previous stage are weighted in views of selecting the most important terms as representatives of the original text.

The use of frequency as a characteristic in AGTS has demonstrated being useful. The frequency of the term was used for the first time in extractive AGTS in the 1950's decade (Luhn, 1957). Later researches have utilized frequency in their methods to identify important terms in their works. More recently, SumBasic algorithm utilized the frequency of the term as a part of a context-sensitive approach to identify important sentences at the time that it reduces information redundancy (Nenkova, Vanderwende, 2005).

The weighting proposed by (D'Avanzo, et al., 2007) is based on a combination of term frequency (tf) by the inverse document frequency (idf) and its first appearance. This is to say, the distance of the candidate term (or specifically, the keyphrases are utilized as terms) from the beginning of the document in which they appear. Inverse frequency is defined as the number of documents in which the term appears divided between the number of documents the collection has; making the terms more important if they appear in few documents, which allows characterizing each document as best as possible. For example, the prepositions ("the", "in", etc.) or the connectors of conjunctions or disjunctions ("and", "or"), even though they appear very frequently in the text with tf, with idf they would not be very important for they virtually appear in all documents.

(Nenkova, Passonneau, 2004), (Passonneau, et al., 2007), (Nenkova, 2006) consider term weighting by using a pyramid schema –a procedure specifically designed to comparatively analyze the content of several texts. The idea of this schema consists in calculating the presence of each term in all the collection documents; the more documents contain the term, the more important the term is, therefore it shall be included in the summary. Although Nenkova's schema is applied to multiple document AGTS, it would be possible to apply it to a single document if instead of

considering a collection of documents, a collection of sentences from a single document is considered.

Wei et al., 2006) derivate the relevance of a term from an ontology built with the analysis of formal concepts. (Song, Han, Rim, 2004) basically weight a word on the basis of the number of lexical connections, as the semantical associations expressed in a thesaurus, the word has with its neighboring words; and with this, the most frequent words will be weighted as the most relevant.

(Mihalcea, 2006) presents a similar idea in the shape of a network, a formalism based upon a graph: the words with a closer relationship with a larger number of "important" words will be more important on their own; the importance is estimated in a recursive manner similar to PageRank (Brin, Page, 2012), an algorithm used by Google to weight the importance of web pages. The idea is that a sentence is important if it is related to many important sentences, in which this relation can be understood as the overlapping of the lexical contents of the sentences (Mihalcea, 2006). The two methods presented in (Mihalcea, 2006) for AGTS are some of the best results reported in the literature and which we will compare our proposed method with.

As it is noticed in the previous works, the relevance of each term is determined by the frequency of a simple count of their appearance in the same document or in the collection of documents; or by the frequency of their structural or semantic relation of the term context with other terms.

It is worth pointing out that no works have been found in which MFSs are used for AGTS, so there is no schema for the initial weighting. However, it would be possible to utilize the frequency and length of each MFS found in the document as two probable variables that might reflect the importance of each term.

II.2.1.3 Sentence weighting

In (Cristea, 2005) the weighting of a sentence is performed in function of its proximity to the central idea of the text, which is determined by analyzing the discourse structure.

However, the techniques that try to analyze the structure of the text imply an excessively sophisticated and costly linguistic processing. By contrast, the largest part of the methods described in the literature nowadays represent the text and its phrases as a bag of simple characteristics using statistical processing with no attempt to "understand" the text at all.

A very old and simple heuristic for sentence weighting does not imply any term whatsoever: it only assigns a heavier weight to the first

sentences of the text. Texts of some genres –such as news reports from newspapers or scientific texts– are designed following this heuristic. For example, any scientific article contains an abstract at the beginning. This simple heuristic has shown to be very difficult to overcome by other automatic methods, so it has been taken as the baseline in (DUC, 2014) for a new AGTS method. This is to say, baseline is the minimal quality to surpass that new AGTS methods must have, as they will be making a "more intelligent" processing of the text than only taking the first sentences of the document. It is worth pointing out that in the competitions of the Document Understanding Conference (DUC, 2014), only five systems performed over this baseline, which does not undermine the other systems because this line is specific for the genre. For example, this would not work in official documents, email messages, web pages or literary novels, as the position of the sentence would not be available for term-based methods.

Another of the possible approaches is Relative Utility proposed by (Radev, Tam, Erkan, 2003). In this approach all the input sentences are graded at a scale from 0 to 10, according to their aptitude to be included in a summary. Moreover, the sentences that contain similar information are explicitly marked, so the computer assessment might punish or reward the redundancy of informatively equivalent sentences. This method can only be applicable to extractive systems that directly select sentences from the input, but not to abstractive systems.

(Verma, Chen, Lu, 2007) utilizes the knowledge of ontologies to weight sentences using statistical data of sentences, as well as syntactical analysis. The disadvantage of this proposal is that it was made for a single domain in particular, besides it needs a syntactical analyzer.

II.2.1.4 Sentence selection

Supervised learning methods consider the selection of sentences a classification task. To do so, a classifier is trained using a collection of supplied documents with existing summaries. The previously selected terms with their respective weights are considered characteristics of the sentence (Villatoro-Tello, et al., 2006); some characteristics both lexical and non-lexical have even been used (Kupiec, Pedersen, Chen, 1995), (Neto, Freitas, Kaestner, 2002), (Chuang, Yang, 2004). In Kupiec's work (Kupiec, et al., 1995), the following characteristics were proposed: position of the sentence in the document, sentence length, presence of keyphrases in the sentence and the overlapping of the words in the sentence with the document title.

More recent works (Chuang, Yang, 2004), (Neto, Freitas, Kaestner, 2002) extend these characteristics incorporating information on the occurrence of proper pronouns and the presence of anaphoras. The "heuristically motivated" characteristics allow extracting better summaries. However, they have the very important disadvantage of being closely linked to a specific domain. This condition implies that the change from one domain to another makes it necessary to redefine or even suppress some of the characteristics. For example, the keyphrases that are particular for each domain would require modification, while the overlapping of words with the title would not have a meaning in every case, so they might be suppressed.

Chali and Kolla (Chali, Kolla, 2004), (Kolla, Chali, 2005) presented a work for multiple document AGTS using lexical chains to select sentences. In (Chali, Kolla, 2004), the mechanism to score only considers the times the word appears in a sentence and in a segment (a segment is comparable with single documents within a single-topic collection), no additional information, such as n-grams, is used. The obtained results indicate that the score of a sentence based on simple lexical chains does not improve sufficiently. As a consequence, weighting was changed in (Filippova, et al., 2007) by adding various scores. An added score was the number of chains that pass through a sentence (chain score) and the other score is based on n-grams (bigrams and trigrams). Each occurrence of a chain and bigram increases the score in 1, while a trigram, in 2. The global score is calculated for each phrase and then all the sentences are weighted for the final extraction. Using the number of chains that pass through a sentence yields higher scores for longer sentences. This takes place because longer sentences also have a higher probability of containing more information —especially, if several chains pass through them.

The approach in (Sek, 2002) bases upon weighting calculated as follows. In the first place, the values of tf x idf are calculated for all the substantives in the document, with the exception of stop words (which lack a proper meaning and are shown in annex A). Then, for each document the addition of all the tf x idf of the substantives is calculated. The importance value of a sentence is calculated by means of adding all the tf x idf values of sentences that contain substantives. The value of resulting importance for each sentence is thus obtained.

The extractive approaches for AGTS usually follow a model to weight sentences based on a set of characteristics; the sentences with the highest scores are extracted to produce the summary. When frequency is used as the only characteristic, the elements of the term are counted and then each sentence will have a score based on the calculation of the frequency of each element in the sentence. A key problem in the gene-

ration of summaries is the reduction of redundancy. Each new sentence in the summary should add new information instead of repeating that already included. The use of the most frequent terms will probably end up as the same information repeated several times. In the work SumBasic frequency method (Nenkova, Vanderwende, 2005), a probability distribution model is generated in the first place and each term is utilized to select the sentences; the probabilities of the term are reduced, so the terms with the lowest probability have a better opportunity to select sentences with new information content. This approach is called context sensitive, as the AGTS system considers the sentences already in the summary before adding a new sentence to the summary. This is also related to the idea of finding maximal marginal relevance, where marginal relevance is defined as the search for relevant sentences that contain minimal similitude with the previously selected sentences (Carbonell, Goldstein, 1998).

The frequency distribution of FreqDist algorithm uses a context-sensitive approach to weight the sentences that are based on the frequency distribution model, instead of a probability distribution model (Reeve, Han, 2007). The substantiation of the frequency distribution approach is that the distribution of the frequencies of the terms or concepts in the original text has to appear in the generated summary as closely as possible to the original text. This is to say, the frequency distribution models of the original texts and their corresponding summaries have to be as similar as possible. There are two stages in the algorithm: initialization and summary generation. In the initialization stage, the elements of the units (terms or concepts) of the original text are counted to produce a frequency distribution model of the text and a series of sentences from the test is created, which is called sentence series. A frequency distribution model of the summary is created from the elements of the terms found in the original text. The count of the frequency distribution model of the summary is initially set to zero to indicate an empty summary. In the summary generation stage, new sentences are assessed and then selected to be included in the summary. The identification of the next sentence to be integrated in the summary is carried out by means of searching the sentence that better aligns the frequency distribution of the summary so far generated with the frequency distribution of the original text. For each sentence in the sentence series, the sentence is added to the candidate summary in order to find out how much it contributes to the candidate summary. To determine the contribution of the sentence, the frequency distribution of the candidate summary is compared to the frequency distribution of the original text. This comparison generates a similitude score, which is assigned to the sentence as the sentence score. Once all the sentences (in the sentence series) have been assessed by their contribu-

tion to the synthesis of candidates, the highest scored sentence is added to the summary and removed from the sentence series. The process to select sentences is iterative and repeated until the desired length of the summary is reached.

Most of the current methods are purely heuristic: they do not use any learning, but directly establish the procedure utilized for term selection, terms and/or sentence weighting (since the selection in most of the cases consists in the selection of the best weighted sentences).

II.2.2 Automatic generation of abstractive summaries

Abstractive AGTS methods use the extraction of information, ontological information, fusion and comprehension information. Automatically generated abstract summaries (abstractive summaries) move the field of purely extractive AGTS methods to the generation of abstract summaries that contain sentences which were not in any of the input documents and that can synthesize information by means of a number of sources. An abstractive summary contains at least some sentences (or phrases) that did not exist in the original document. Of course, actual abstraction consists in taking the process one step beyond. Abstraction implies the recognition of a set of extracted passages, which as a set become something new, something that is not explicitly mentioned in the source and then is replaced in the summary with new concepts (ideally, more concise). The requirement that the new material that is not in the text explicitly means that the system must have access to some sort of external information, as an ontology or a knowledge base in which it is capable of making combinatory inferences.

Various abstractive methods have been developed for AGTS. For instance, the techniques of sentence fusion (Daume, Marcu, 2004), (Barzilay, 2003), (Barzilay, McKeown, 2005), information fusion (Barzilay, et al., 1999), sentence compression (Vandeghinste, Pan, 2004), (Madnani, et al., 2007), etc.

II.3 Assessment of automatic summaries

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) was proposed by Lin and Hovy Lin y Hovy (Lin, Hovy, 2004), (Lin, Och, 2004), (Lin, Och, 2004a). This system calculates the quality of an automatically generated summary comparing it with the summaries produced by human beings. In short, it counts the different common units, such as

word sequences, word pairs and n-grams in the summary to assess (the computer-generated one) and the ideal summaries created by humans. ROUGE includes a number of automatic assessment measures.

ROUGE-N (n-grams co-occurrence):

It expresses the coverage or recall of *n*-grams between a candidate summary and a set of reference summaries. It is calculated as follows:

$$ROUGE-N = \frac{\sum_{O \in \{ReferenceSummaries\}} \sum_{gram_n \in O} count_{coincidence}(gram_n)}{\sum_{O \in \{ReferenceSummaries\}} \sum_{gram_n \in O} count(gram_n)},$$

Where n is the length of the n-gram and count $_{coincidence}$ (gram n) is the maximal number of n-grams that co-occur in the candidate summary and in the set of reference summaries.

ROUGE-L (longest subsequence): a sequence S = (s1, s2, ..., sn) is a subsequence of another sequence X = (x1, x2, ..., xm), if there exists a strict sequence on the increase (i1, i2 , ..., ik) of the indexes of X so that for every j = 1, 2, ..., k, there is xij = sj. Given two sequences X and Y, the longest common subsequence (LCS) of X and Y is a common subsequence with maximum length. When LCS is applied to the assessment of summaries, a sentence of the summary is seen as a sequence of words. Intuitively, the LCS of two sentences is the most similar of the two summaries X and Y, where X is m in length and Y n in length, supposing X is a sentence from the summary and Y is a sentence from the candidate summary.

ROUGE-W (longest weighted subsequence): given two sequences X and Y, LCS is called weighted if the length is calculated using the weighting function. For further details on the weighting function, see (Lin, Hovy, 2004).

ROUGE-S (noncontiguous bigram co-occurrence): a noncontiguous bigram is any pair of words in the order of the sentence, which allows for an arbitrary number of spaces. The co-occurrence of noncontiguous bigrams statistically measures the coverage of noncontiguous bigrams between the candidate summary and the set of reference summaries.

It was shown in Lin's work (Lin, Hovy, 2003) that this sort of measures can be applied to assess the quality of automatically generated summaries, as they managed a 95% correlation between human judgments.

For each of the ROUGE measures (ROUGE-N, ROUGE-L, ROUGE-W, etc.), Recall, Precision and F-measure are calculated as follows (see example in Annex B).

Precision (P): it shows the quantity of good sentences extracted by the system:

$$P = \frac{\#(correct\ sentences)}{\#(correct\ sentences + incorrect\ sentences)}$$

Recall: (R): it shows the amount of correct sentences the system forgot:

$$R = \frac{\#(correct\ sentences)}{\#(correct\ sentences + unextracted\ sentences)}$$

F-measure (F):

$$F = \frac{2PR}{P+R},$$

Where, correct sentences are the number of sentences extracted by the system and humans; incorrect sentences are the number of sentences extracted by the system, but not by humans; and not extracted sentences are the number of sentences extracted by humans, but not by the system.

II.4 Assessment of commercial tools and state-of-the-art methods to generate individual extractive summaries.

In this section we assess the quality of seven commercial tools and three state-of-the-art methods for AGTS using DUC-2002 news item collection and ROUGE comparison tool (presented in the previous section).

II.4.1 Description of commercial tools

Commercial tools can be classified in installable and online, according to their execution. Svhoong Summarizer (Svhoong, 2013), Pertinence Summarizer (Pertinence, 2010), Tool4noobs Summarizer (Tool4noobs, 2013) and Open Text Summarize (Ots, 2013); and other commercial two: Copernic Summarizer and Microsoft Office Word in its 2003[1] and 2007[2] versions. These tools are described below.

Shvoong (Shvoong, 2013) was founded in 2005 by Avi Shaked and Avner Avrahami. Shvoong is a tool that allows generating automatic summaries in 21 different languages (Czech, Dutch, Danish, English, Finnish, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Malayan, Norwegian, Polish, Portuguese, Romanian, Russian, Spanish and Turkish). Un-

[1] Microsoft ® Office Word 2003. SP3 Part of Microsoft Office Professional Edition 2003 Copyright © 1983-2003 Microsoft Corporation.

[2] Microsoft ® Office Word 2007. Part of Microsoft Office Professional 2007 © 2006 Microsoft Corporation.

like other tools, Shvoong does not return the summary as such, but underlines the text that it considers most important in the original document.

Pertinence Summarizer (Pertinence, 2010) belongs to the product range developed with KENiA© technology (based on knowledge extraction and notification architecture) developed by French company Pertinence Mining. Pertinence is an online tool that allows generating summaries in 12 languages (German, English, Arabic, Chinese, Korean, Spanish, French, Italian, Japanese, Portuguese, Russian and Dutch) of text documents in various formats (html, pdf, doc, rtf and txt).

Tools4Noobs (Tool4noobs, 2013) is an online tool that allows generating summaries from 1 to 100% of the original text. The generation of a summary in Tools4Noobs has three stages: extraction of sentences, identification of keywords in a text counting the relevance of each word, and identification of sentences according to the identified keywords.

Open Text Summariser (Ots, 2013) is an open code application to summarize texts; it can be downloaded from the Internet free of charge; however, it has an online interface as well. OTS generates automatic summaries at various percentages and can generate summaries in 37 languages.

Copernic Summarizer is a software program exclusively designed for AGTS; it works with four languages (English, German, French and Spanish) (Copernic, 2013).

Microsoft Office Word is an office automation suite to process and edit text that includes an AGTS option.

II.4.2 Brief description of state-of-the-art methods

Some of the state-of-the-art methods are: graph-based weighting TextRank (Mihalcea, 2004a). It was also compared with the heuristic methods called Baseline (see section II.2.1.3) and Baseline: random (see section V.3.3); which are references to measure the advance of the state-of-the-art methods. Baseline is a heuristic that consists in taking the first sentences of the text to generate the summary (Villatoro-Tello, et al., 2006). Baseline: random is another heuristic whose functioning consists in taking some text sentences at random. So any method that behaves as baseline: random would not have a reason to exist.

II.4.3 Assessment configuration

In order to be able to undertake the comparison between AGTS online tools and state-of-the-art methods, we will use the Document Understand-

ing Conference [Duc] collection. DUC-2002 was created by the National Institute of Standards and Technology (NITS) to be utilized by researchers in the area of summary generation; it comprises 567 news items in English of various lengths and on diverse topics. Each DUC-2002 item has two 100-word summaries produced by two human experts.

To assess the summaries automatically generated by commercial tools we will use ROUGE 1.5.5., described in the previous section. The assessment consists in estimating the similitude of the automatically generated summaries with those written by human experts. The summaries generated with the installable commercial tools and with the online tools were generated with at least 100 words, so each tool was analyzed to meet the minimal length of the automatic summary.

II.4.4 Assessment of online and installable commercial tools of AGTS for a single document

Four online and two installable tools were compared. In figure II.1, it is noticed that Shvoong and OTS online tools obtained better results than Microsoft Office Word tools. The other online tools, tools4Noons and Pertinence, were the ones with the poorest results, while the best result was obtained by the installable tool Copernic Summarizer.

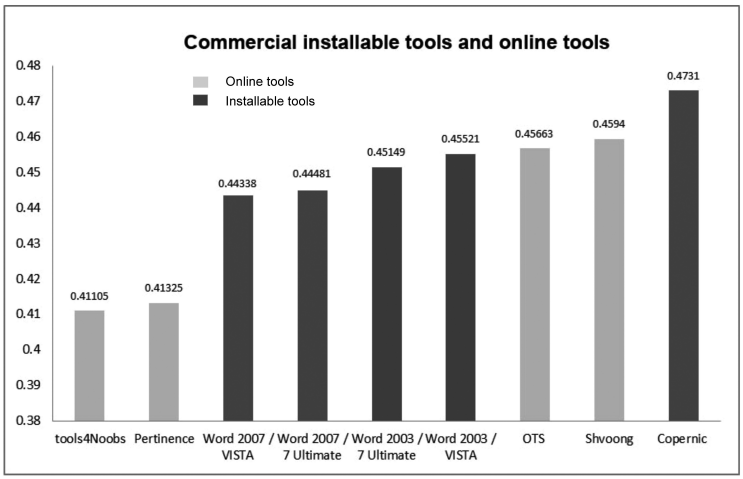


Figure II.1 Comparison of installable and online commercial tools.

The results obtained with Microsoft Office Word on Windows 7 Ultimate did not surpass the summaries in versions 2003 and 2007 on

Windows Vista. This way, the value considered for the comparison of the installable tool Microsoft Office Word with the state-of-the-art methods and online tools will be the one obtained with the version 2003 on Windows Vista.

II.4.5 Assessment of commercial tools and state-of-the-art methods

In views of acknowledging the advancement commercial tools have had in comparison with the state-of-the-art methods, the previous results were included with other seven state-of-the-art methods in Figure II.2.

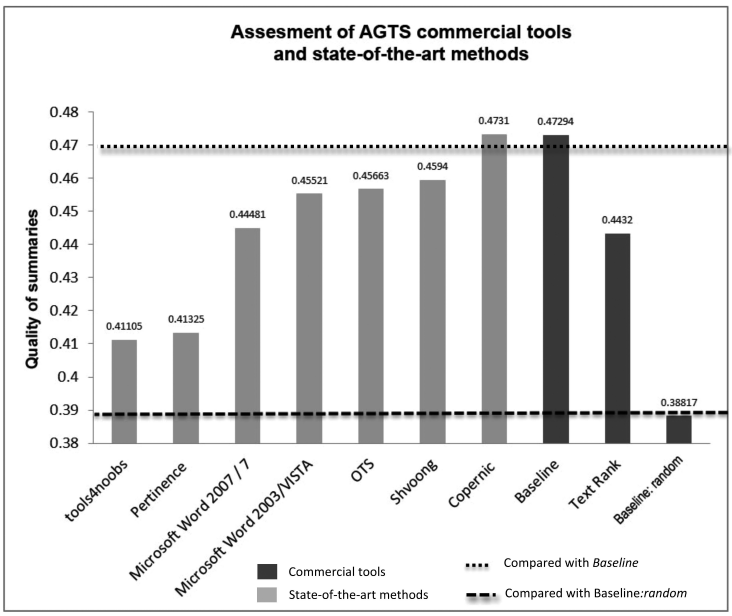


Figure II.2 Assessment of AGTS commercial tools and state-of-the-art methods.

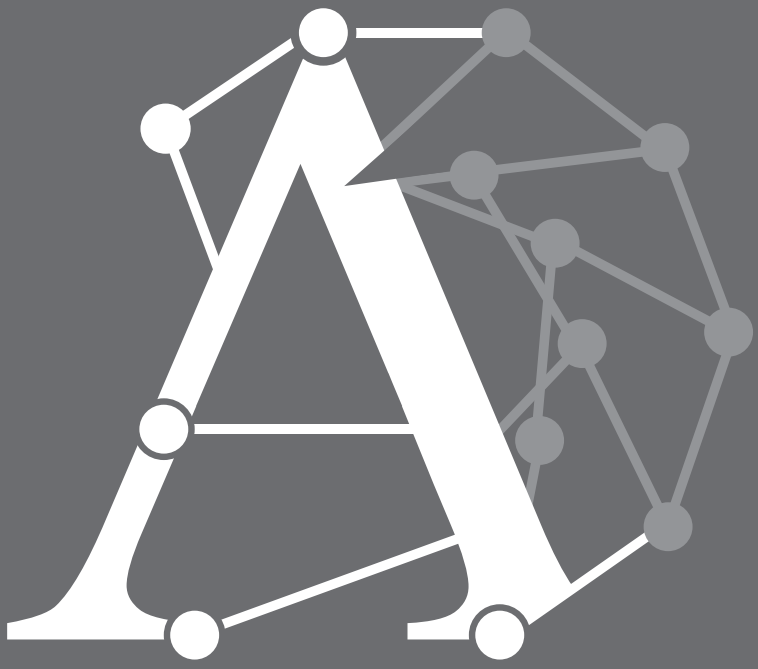
In Figure II.2 one observes that the results of commercial tools are below some of the results of the state-of-the-art methods. This is to say, the state-of-the-art methods have good quality.

One of the heuristics to be overcome by installable and online commercial tools, as well as the methods proposed in the state of the art is Baseline. As it is seen in Figure II.2 only Copernic Summarizer (installable commercial tool) overcomes this heuristic.

It is worth mentioning that despite online tools do not overcome Baseline, some of them such as OTS and Shvoong are over the results

yielded by Microsoft Office Word and the state-of-the-art methods TextRank and SFM (K-best).

In particular, it was found that out of the four AGTS online tools, the best was Shvoong Summarizer. However, none of the four AGTS online tools overcame Baseline. As a result of the comparison of all the commercial tools, both installable and online, it was noticed that Copernic Summarizer is still the only AGTS commercial tool that overcomes Baseline.



CHAPTER III.
THEORETICAL FRAMEWORK

In the previous chapter we analyzed by stage the various combinations utilized by state-of-the-art AGTS methods. In this chapter we will present the basic definitions of the methods and models that will be part of the methods proposed in the following chapter. In section III.1, we will review the methods utilized to preprocess the text that are commonly followed by NLP applications. In section III.2, we present definitions and examples of the terms that can be selected independently from language and domain. In particular, we present the formal definition of maximal frequent sequence, since the proposed methods will use this sort of term (see section III.2.3). In section III.3, we present the weighting of terms that are commonly used in IR and AGTS. In section III.4, the three types of methods proposed for the weighting and selection of sentences are outlined. In section III.5, we introduce the genetic algorithms, particularly, the basic one, a representation of the population fitness function and genetic operators.

III.1 Text preprocessing

The preprocessing stage depends on how efficient the representation of a text is, thereby it is important in the AGTS area as it allows increasing the quality of the obtained summary. The proposed methods contemplate the stage of preprocessing. By and large, this stage will include only two steps: the elimination of stop words (see annex A) and the lemmatization of words by means of stemming.

III.1.1 Suppression of stop words

When the preprocessing of a text is carried out, an intermediate representation of it is obtained. One of the stages of preprocessing consists in suppressing stop words from the text. There is a set of stop words in every language, common to all domains, that are easy to identify, for instance articles, prepositions, conjunctions, etc., nevertheless they can be verbs, adjectives and adverbs; these words are considered empty and commonly eliminated.

Words that are very frequent in documents in a determinate collection are not good discriminants. As a matter of fact, it is considered that a word that appears in at least 80% of the documents of a certain collection is useless for the purpose of information retrieval. For AGTS, it is also indispensable to suppress this sort of terms, since, as it will be seen in next

section, they present a very high frequency; which suggests they are very important terms which would produce summaries with information gaps.

Further in the text we will carry out a process of stop-word extraction in the documents in views of reducing the content of text to the most specific expressions (we call them multiword descriptions), which only contain the words that are useful and significant to generate automatic summaries.

Although the list of stop words depends on language (for an example of stop words in English see annex A), the list of words is not so large; around 200 words and there are several lists published in various languages, this way, the suppression of stop words is considered to be scarcely dependent on domain and language.

III.1.2. Arbitrary word lemmatization (Stemming)

With the objective of being able to relate words that mean the same, but which are written slightly different, for instance: "*perro*", "*perros*", "*perra*", "*perrito*", "*perrotes*",[1TN] etc., lemmatization or arbitrary normalization algorithms (stemming) can be applied; these mainly truncate affixes, prefixes and postfixes of the words with the objective of trying to obtain a "root" of the word, which for the previous example may be "*perr*" [dog]. It is supposed that two words with the same root represent the same concept. Basically, the stemming process is carried out to reduce a common part of the word, called stem, to the minimal; this is the part of the word that remains after suppressing its affixes, prefixes and suffixes. The stemming algorithms try to simulate lemmatizers based on dictionaries, which search for the linguistic roots of each of the possible forms of a word in dictionaries. However, for the objectives of this research, the stemming technique is desirable, as it requires the least linguistic resources. Another advantage of stemming is that it can work with non-previously identified or incorrectly spelt words. This will simplify the representations of documents by means of the aforementioned models.

The first stemming algorithm was developed for the English language and then it was adapted to Spanish. Porter's Algorithm (Porter, 1980) is the most utilized in the English language. There are also algorithms for other languages such as French, Dutch, Greek and Latin. In general, these algorithms are based on a simple set of rules that truncate words in order to obtain a common root (Baeza-Yates, et al., 1999).

[1] [TN] Spanish language words which in English mean: dog, dogs, bitch, puppy, big dogs.

III.2. Term selection models

Since they virtually do not depend on language and domain, the term selection models considered in this research are bag of words (Salton, Wong, Yang, 1975), n-grams (Villatoro-Tello, et al., 2006) and MFSs (Ahonen-Myka, 1999).

III.2.1 Bag of words

The representation with bag of words consists in obtaining all the different words that appear in the text. Later on, the document will be represented by the set of words; with the sequential order of those words lost. The bag of words model is easy to extract, however by considering a single word polysemy increases due to the loss of context of the term itself. Another problem brought along by this model is that there are too many terms, even in a small text.

For example, for the collection of five documents (in Spanish in the original) in figure III.1, there will be 19 terms: "*the*", "government", "*of*", "*Egypt*", "*protects*", "*the**"(*las*, feminine plural), "*pyramids*", "*are*", "*a*", "*heritage*", "*cultural*", [they] "*were*", "*built*", "*by*", "*the***" (*los*, masculine plural), "pharaohs", "tombs", "*for*" and "*good*"

F. The government of Egypt protects the pyramids*
G. The pyramids of Egypt are a cultural heritage
*H. Pyramids were built by the** the pharaohs*
I. The pyramids of Egypt were tombs for the pharaohs of Egypt
J. A good government protect its cultural heritage

Figure III.1 Example of 5 sentences of an arbitrary text.

III.2.2 N-grams

Facing the problem of loss of context bag of words has, contiguous word sequences of a predetermined *n* size have been utilized; these are known as n-grams. The n-gram model follows the same principle as the model based on bag of words, this is to say, the difference is that the size of the gram, this is to say n, is the number of consecutive elements that contain the element. As a matter of fact, the bag of words model can be represented as 1-grams. For example, if n is equal to 2, the term will contain 2 consecutive words of the original text (also called bigrams).

Considering the collection of documents in figure III.1, the following 25 bigrams will be obtained: "the government", "government of", "of Egypt", "Egypt protects", "protects the", "the pyramids", "pyramids of", "Egypt are", "are a", "a [cultural] heritage", "cultural heritage", "pyramids were", "were built", "built by", "by the", "the pharaohs", "Egypt were", "were tombs", "tombs for", "for the", "pharaohs of", "a good", "good government", "government protects" and "protects its".

As in the bag of words model, the extracted terms do not fully preserve the order in which they appear in the text. Moreover, there is another inconvenient common to both models, there is a considerable amount of different characteristics to assess; this supposes an enormous resource expense to handle such amount of information.

Even if the bigram or trigram model normally improves the semantic representation of the terms for various NLP tasks, it is not clear what the ideal size for each application, domain or language would be.

III.2.3 Maximal Frequent Sequences (MFSs)

Trying to solve the problems of sequential order and dimensionality of the models, MFSs have been proposed as a model to represent the text (Ahonen-Myka, 1999).

A frequent sequence (FS) is sequence of words, in which these appear in the same sequential order and repeatedly. An FS is called maximal if it is not contained in another FS. The MFS model finds the number of times that the FS will repeat over the text to be considered frequent; this number is called frequency threshold.

It is important to bear in mind the large amount of frequent sequences there might be in a small collection of documents. For instance, considering the five sentences in figure III.1 with a frequency threshold 2, the following 18 contiguous FSs can be generated.

"government"	"[they] were"
"of"	"the"[2] [*los*]
"Egypt"	"pharaohs"
"protects"	"of Egypt"
"the"[3] [*las*]	"the pyramids"
"pyramids"	"pyramids of"
"a"	"Cultural heritage"
"heritage"	"the pharaohs"
"cultural"	"the pyramids of Egypt"

[2] "*Los*", masculine plural.

[3] "*Las*", feminine plural.

By observing the previous list, it is noticed why it is important to only take the MFSs.

A formal definition:

An *S* sequence, denoted by $<s_1,s_2, \ldots,s_k>$, is an ordered list of *k* elements. A sequence of *k* length is called k~sequence.

Be $P=<p_1, p_2, \ldots, p_t>$ and $S= <s_1,s_2, \ldots,s_m>$ sequences, P is a subsequence of S, with GAP=0, denoted as $P \subseteq_0 S$ if there is an integer $i \geq 1$ such that

$$p_1= s_i, p_2 = s_{i+1}, p_3 = s_{i+2}, \ldots, p_t= s_{i+(t-1)}$$

A *W* sentence in a *D* document can be considered as a sequence of words, also denoted as

$$<w_1, w_2, \ldots, w_t>$$

The frequency of an *S* sequence in a collection of sentences $\{W_1, W_2, \ldots, W_j\}$, considered sequences, is denoted by S_f or $<s_1,s_2, \ldots,s_t>_f$ that is the number of sentences in which *S* appears at least once. This is $S_f = |\{W_i | S \subseteq_0 W_i\}|$.

Given a user-defined threshold (β), an *S* sequence is frequent if $S_f \geq \beta$. An S frequent sequence is maximal if it is not a subsequence of any other frequent sentence.

A collection of sentences $\{W_1, W_2, \ldots, W_j\}$ composes a D document when $\{W_1, W_2, \ldots, W_j\} \in D$

Given a document, all the maximal frequent sentences have to be extracted considering GAP and threshold β.

For instance, the document presented in figure III.1 is composed of five sentences from which with β =2 and GAP=0, the following MFSs are found: "*A*", "government", "were", "*protects*", "cultural heritage", "the pharaohs" and "the pyramids of Egypt".

Using the same document in figure III.1, with β =2 and GAP=2, the following MFSs are found: "a", "government protects", "the Egypt", "cultural heritage", "the pyramids of Egypt", "*the of Egypt*", "the pyramids of Egypt" and "the pyramids were the pharaohs". More examples of MFSs from DUC-2000 collection can be found in Annex C.

III.3 Term weighting

Term weighting consists in assigning a value for each term that reflects the importance of this term in relation to other terms in the document. As it was exposed in the previous chapter, term weightings independent

from language are based on the frequency of the term in the document. Now the weighting of the most common terms is described.

Boolean weighting; it is the easiest way to weight a term. The value of 1 is assigned, if appears in sentence and value 0, in other case.

$$p_i(t_j)=\begin{cases}0, & in \quad other\ case\\ 0, & if \quad it\ appears\end{cases} \tag{3.1}$$

Where t_j is the frequency of j term in sentence p_i. In table III.1, the Boolean weighting that corresponds to sentences in figure III.1 is shown, using MFSs as terms with β =2 and GAP=0.

Table III.1 Boolean weighting based on MFS with β =2 and GAP=0 for sentences in figure III.1

MFS \ Document	A	B	C	D	E
A	0	1	0	0	1
Were	0	0	1	1	0
Government	1	0	0	0	1
Protects	1	0	0	0	1
Cultural heritage	0	1	0	0	1
The pharaohs	0	0	1	1	0
The pyramids of Egypt	0	1	0	1	0

Term frequency (*tf*) was proposed in (Luhn, 1957). This weighting takes into account that a term, which is frequently repeated in a sentence, can reflect the content of a sentence better than a term which is repeated fewer times. Therefore, *tf* weighting assigns a heavier weight to the terms with the highest frequency and consists in assessing the times the word appears in the sentence.

$$p_i(t_j)= f_{ij} \tag{3.2}$$

Where, f_{ij} is the frequency of j term in sentence i. In table III.2 we show the frequency weighting corresponding to sentences in figure III.1, using MFSs as terms with β =2 and GAP=0.

Table III.2 frequency weighting based on MFSs with β =2 and GAP=0 for sentences in figure III.1

SFMs \ Document	A	B	C	D	E
A	0	1	0	0	1
Were	0	0	1	1	0
Government	1	0	0	0	1
Protects	1	0	0	0	1
Cultural heritage	0	1	0	0	1
The pharaohs	0	0	1	1	0
The pyramids of Egypt	0	1	0	1	0

Inverse document frequency (*idf*) was proposed in (Salton, Buckley, 1988). It considers that a very frequent term which appears in a number of documents is less useful than a term that appears less frequently but in few documents. What is assessed is the distribution of the terms in a document; the inverse frequency of the document is defined as:

$$p_i(t_j) = \log(\frac{N}{n_j}) \tag{3.3}$$

Where f_{ij} is the frequency of *j* term in document *i*; *N* is the number of documents in the collection; n*j* is the number of documents in which *j* term appears.

It is worth mentioning that this measure can be applied in AGTS to a single document when it is considered that there is a collection of sentences instead of one of documents.

In table III.3, we show the inverse document weighting corresponding to the sentences (taken as documents) in figure III.1, using MFSs as terms with β =2 and GAP=0.

Table III.3 Inverse document weighting of the document based on MFSs with β =2 and GAP=0 for sentences in figure III.1

SFMs \ Document	A	B	C	D	E
A	0	0.398	0	0	0.398
Were	0	0	0.398	0.398	0

Government	0.398	0	0	0	0.398
Protects	0.398	0	0	0	0.398
Cultural heritage	0	0.398	0	0	0.398
The pharaohs	0	0	0.398	0.398	0
The pyramids of Egypt	0	0.398	0	0.398	0

tf-idf weighting. It is common that term frequency (tf) and inverse document frequency (idf) are used together in views of finding the relevance of each term, considering both the importance the term has in the collection and its importance in the document (Salton, Buckley, 1988). This combination is known as *tf x idf* weighting and it consists in multiplying the frequency of the term in relation to the document by the inverse frequency of the document in which this term appears.

$$p_i(t_j) = f_{ij} \text{ x} \log\left(\frac{N}{n_j}\right) \tag{3.4}$$

In this method it is important to take into account that we are working with a single document, so N will be the number of sentences and n_j, the number of sentences in which the term appears. Following the example of the collection of documents in figure III.1, with *tf-idf* weighting the same weights as in table III.3 will be obtained; this is because the collection is very small so it is difficult to find a MFS in more than a sentence.

Weighting according to length. As it was shown in the previous section, terms with multiword descriptions enhance their semantical content, so it is intended to integrate the terms into several words that offer a meaning. MFSs, unlike the terms of bag of words and *n*-grams, vary the quantity of words, so in this research we propose to use the length of MFS as a measure of the relevance of the term. This is to say:

$$p_i(t_j) = |t_j| \tag{3.5}$$

In table III.4, we show the length weighting of the sequence corresponding to the document collection in figure III.1, using MFSs as terms with β =2 and GAP=0.

Table III.4 Length weighting based on MFSs with β =2 and GAP=0 for the collection of documents in figure III.1

SFMs \ Document	A	B	C	D	E
A	0	1	0	0	1
Were	0	0	1	1	0
Government	1	0	0	0	1
Protects	1	0	0	0	1
Cultural heritage	0	2	0	0	2
The pharaohs	0	0	2	2	0
The pyramids of Egypt	0	4	0	4	0

III.4 Weighting and selection of sentences

Sentence weighting has as a final step the selection of sentences; thereby, these two stages are normally worked together by various algorithms, for instance clustering (see, section IV.5), genetic (García-Hernández, Ledeneva, 2013) and graph weighting algorithms (Mihalcea, 2004a). As commented in the state-of-the-art section, TextRank (Mihalcea, 2006) (an adaption of PageRank, which weights web sites to use them in Google) has shown to be a good method to weight and select the sentences that will be part of the summary

III.4.1 TextRank algorithm

In this section the functioning of TextRank algorithm is explained. In order to carry out the application of weighting algorithms based on graphs for texts in natural language, a graph to represent the text is built, it interconnects words or other text entities with significant relations. The graphs built this way center on the source text, but they can be broadened with external graphs, such as semantic or associative networks or other similar structures automatically derived from large corpuses.

The nodes or graph vertexes are defined in function of the application, various segments of text can be added, which can have different sizes and characteristics. For instance, words, meanings of words, full sen-

tences, full documents or other. It is worth noticing that the graph nodes do not have to belong to the same category.

The edges or arcs between the graph nodes are defined according to the sort of relation there is between two nodes; for instance: lexical or semantical relations, measures of text cohesion, contextual superposition, the belonging to a word in a sentence, among others.

Algorithm: Regardless of the sort and characteristics of the elements added to the graph, the application of text weighting algorithms is composed of the following main steps:

1. Identify the text segments that best define the task at hand and add them as graph vertexes.
2. Identify the relations that connect such text units and use these relations to create edges between vertexes in the graph. The edges can be directed or non-directed weighted or non-weighted.
3. Apply a weighting algorithm based on graphs to find the relevance in the graph nodes. Iterate the weighting algorithm based on graphs until it converges. Order the edges based on their final score. Utilize the values associated to each edge for weighting/selection decisions.

TextRank general algorithm was defined for AGTS by means of a graph with non-directed weighted edges as follows:

1. The sentences in the original text will be the graph vertexes.
2. The weights of the edges that connect the vertexes will be calculated using cosine similarity between the sentences that such vertexes represent. This way, the sentences are represented as bags of words with the Boolean model; i.e., vertexes X_i and Y_j, corresponding to documents D_i and D_j, will be represented by sets $(x_{1i}, x_{2i},..., x_{mi})$ and $(x_{1j}, x_{2j},...,x_{mj})$, respectively. Cosine similarity between both vectors is calculated as:

$$\cos(D_i, D_j) = \frac{\sum_{l=1}^{m} x_{li} x_{lj}}{\sqrt{\sum_{l=1}^{m} x_{li}^2 \sum_{l=1}^{m} x_{lj}^2}} \tag{3.6}$$

3. To calculate the relevance of the vertexes a recursive iterative process is carried out until all weights converge (i.e., there are no more changes in the weights, or they are below a predefined threshold). At the first iteration, to calculate the first relevance of each node, the sum of its edges, previously calculated, can be used. For later iterations, the weights of the edges are no longer used, but those of the vertexes, which is carried out until convergence. The score of each edge is calculated again

in each iteration on the basis of the new weights that neighboring edges have accumulated. The algorithm concludes when the convergence point for all edges is reached; this means that the error rate for each edge is below a predefined threshold. To do so, PageRank (Brin, Page, 2012) is used, however adapted for non-directed graphs with the following formula.

Given a G directed graph, composed of vertexes *(V)* and edges *(A)*, *G= (V, A)*, where $In(V_i)$ is the set of edges that point at vertex V_i (predecessors) and $Out(V_i)$ is the set of edges that point at vertex V_i (successors), this is $A=\{In(V_i), Out(V_i)\}$. The PageRank algorithm associated to edge V_i is defined as the recursive function $S(V_i)$ that integrates the scores of its predecessors:

$$S(V_i)=(1-d)+d* \sum_{V_j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|} \quad (3.7)$$

Where *d* is the parameter established between 0 and 1.

For example, to calculate cosine similarity between the vertexes that sentences *C* and *D* of the document collection in figure III.1 would make, the representation of table III.5 can be considered, where $D_i = C$ and $D_j = D$.

Table III.5 Representation of weighted bag of words by means of Boolean schema, documents C and D of sentences in figure III.1, employed for the calculation of cosine similarity between C and D

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Doc	the	pyramids	were	built	by	the	pharaohs	of	Egypt	tombs	for
D_i	1	1	1	1	1	1	1	0	0	0	0
D_j	1	1	1	0	0	1	1	1	1	1	1

Where, $\sum_{l=1}^{m} x_{li}x_{lj}= 5$, $\sum_{l=1}^{m} x_{li}^2= 7$, $\sum_{l=1}^{m} x_{lj}^2= 9$, therefore $cos\ (C,D) = 0.63$

Carrying out all the calculations of the edges between the sentences in figure III.1, the graph shown in figure III.2 will be generated in the first iteration. In this graph, the first weights corresponding to the vertexes were calculated; in figure III.2 the initial weighting the sentences have in the document are noticed. In order of importance the sentences would be *B*, *A*, *D*, *C* and *E*. To produce the summary the most weighted sentences are added to cover the number of words or percentage desired by the user.

Once PageRank algorithm is applied to the graph in figure III.2, it will become that in figure III.3. This states that the sentences ordered according to their relevance are *A, B, D, C, E*; where *A* is more relevant than *B* after applying PageRank.

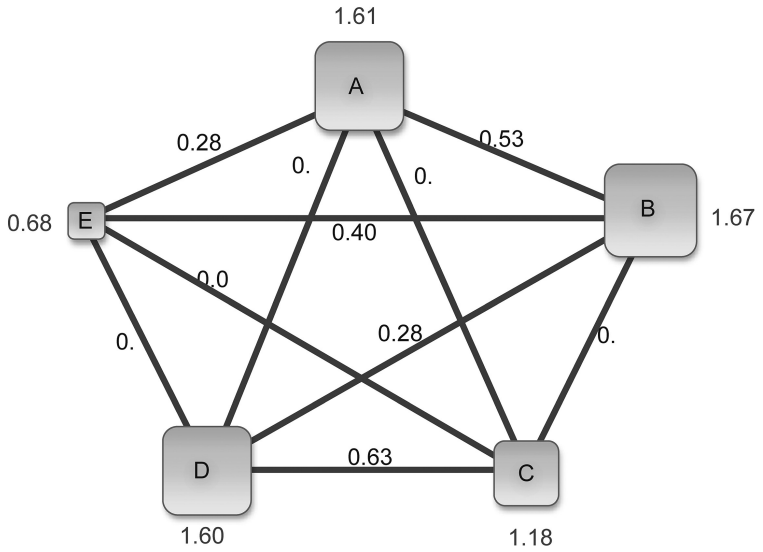


Figure III.2 Representation of the graph used by TextRank (Mihalcea, 2006) to calculate the weighting of sentences in figure III.1. The size of the node represents the initial importance of the sentence in the document.

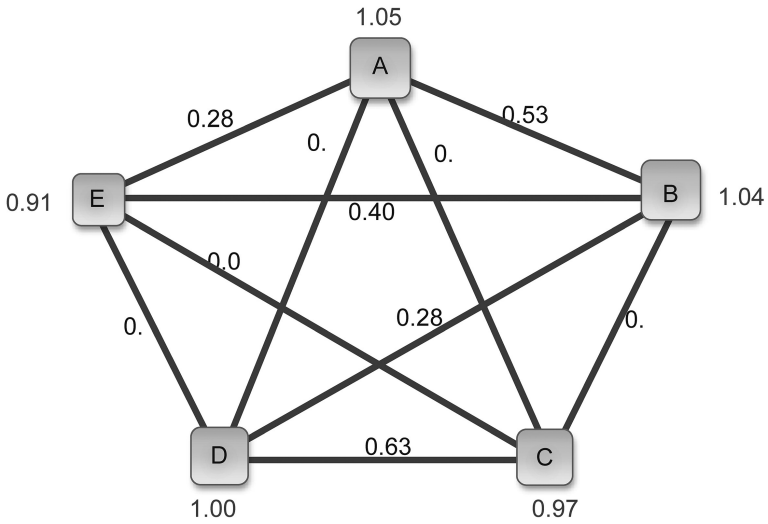


Figure III.3 Representation of the resulting graph by TextRank (Mihalcea, 2006) for sentences in figure III.1. The size of the node represents the final importance of the sentence in the document.

In views of having an approximation both qualitative and quantitative to the quality of the summary generated by TextRank, ten people were asked to order the sentences in figure III.1 from the most to the least important according to their criterion. Moreover, they were also asked to select two sentences as a summary, overlooking whether these agreed with the previously ordered sentences. In order to measure what the order of the selected sentences would be, the most important sentence was given a relevance of 5, the next, 4, and successively. Finally, the relevancies of each sentence were averaged and ordered in a descending manner. The result was: *D* (3.9), *C* (3.9), *B* (3.3), *E* (2), *A* (1.9); where the value in brackets is the average. Considering the previous result as one of the objectives pursued by TextRank, it is noticed that the most important sentences (D and C) were ordered after A and B, however A was the worst sentence under human criterion.

In order to have a qualitative measurement Manhattan distance could be used; it consists of adding the number of positions in which each sentence from TextRank (A, B, D, C, E) differs from the average positions of the surveyed people; since D and C have the same weight, the combination with the lowest measurement will be taken. In this case both combinations *C*, *D*, *B*, *E*, *A* and *D*, *C*, *B*, *E*, *A* yielded 10. For instance, considering *C*, *D*, *B*, *E*, *A* and *A, B, D, C, E* with sentence *A* there is a difference of four positions between *C*, *D*, *B*, *E*, *A* and *A, B, D, C, E*; with sentence B, one, with sentence *C*, three; with D, one; and with E, one; which added yield 10.

As the two most important sentences selected by the people do not necessarily are the sentences that compose the summary (the second sentence can be very similar to the first and not offer information, so another sentence had to be chosen), the survey asked which these two sentences would be. In the survey, the selected sentences with the most votes in descending order were: *D*(7), *B*(6), *C*(3), *E*(2) and A(2). This way, sentences *D* and *B* would clearly be the summary sentences, as they have twice as much the score of C, E and A. As it is seen, humans do not fully agree on that the most important sentences are the ones to be included in the summary. For instance, they agree that the most important sentence is D and that it has to be part of the summary, however according to them, sentence B is in the third place, but it should be part of the summary. It is also noticed that these people agree that sentence A is the worst.

In this case, for AGTS TextRank would select the heaviest weighted sentences that would be A and B. In this case, TextRank and the surveyed agree only on sentence B. It is worth pointing out that the sentence best weighted by TextRank was A, however it was the worst according to the surveyed people.

III.5 Process optimization by means of genetic algorithms

A genetic algorithm (GA) uses the principles of evolution, natural selection and the genetics of biological systems represented in a computer algorithm to simulate the generation of solutions in an evolutionary manner for optimization problems (Goldberg, 1989). Essentially, a GA is an optimization technique that performs a parallel search, stochastic, but directed to make a population evolve turning it more fit.

GAs encode a possible solution to a specific problem in a simple data structure, as the chromosome, and apply crossover operators to these structures so as to preserve critical information. GAs are seen as function optimizers, so there is a broad variety of problems which they have been applied to. The most common application of GAs is the solution of optimization problems, for which they have found efficient and reliable results.

The history of genetic algorithms dates back to the early 1970's decade, when John Holland introduced this concept (Holland, 1975). His objective was to make computers simulate what nature does. Holland was engaged with algorithms that manipulated binary digit strings. Each artificial "chromosome" comprises a number of genes and each gene is represented by 0 or 1:

1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1

Nature has the capability to adapt and learn without being told what to do; this is to say, nature finds good chromosomes blindly, GAs do the same. Two mechanisms link a GA for the problem being solved: encoding and assessment. The GA needs a measure of fitness the individual chromosomes in order to carry out the selection of the best individuals (natural selection according to Darwin (Darwin, 1956)) and their reproduction. According to Darwin, natural selection is the process by means of which the individuals, generally, more fit for their environment manage to reproduce and so inherit through their genes their own strong characteristics, and which by means of breeding with another individual tend to produce better specimens. Nevertheless, in the selection process it is possible that not-so-fit individuals also manage to reproduce, which makes it possible, however unlikely, to obtain better individuals if one or both of the individuals are not so fit. Another of the important processes in evolution according to Darwin, perhaps the most relevant, is the enri-

chment of genetic material by means of the mutation mechanism. Mutation allows descendent individuals to have characteristics that none of the parents previously had, which, in spite of being scarcely frequent, is important to look for new combinations of chromosomes that can generate better individuals; this enables a population to evolve.

III.5.1 Basic genetic algorithm

The basic genetic algorithm consists of the following steps (Negnevitsky, 2005):

Step 1: represent the variables of the domain of the problem as an individual with a fixed-length chromosome; the size of a population of individuals, crossover and mutation probability are chosen.

Step 2: a fitness function is defined to measure the performance or fitness of an individual chromosome in the domain of the problem. The fitness function establishes the base to select the individuals that breed during reproduction.

Step 3: an initial population with a size as follows is generated:
$N: x_1, x_2, ..., x_N$

Step 4: the fitness of each individual is calculated:
$f(x_1), f(x_2), ..., f(x_N)$

Step 5: a pair of breeding individuals from the current population is selected as parents. The parent individuals are selected with a probability related to their fitness.

Step 6: the parents breed to generate a new individual by means of applying a genetic operator.

Step 7: some of the individuals will mutate according to a low probability. The created descendant individuals are placed in a new population.

Step 8: step 5 is repeated until the new population of individuals reaches the number of the initial population.

Step 9: the parents are eliminated from the current population leaving only the descendants.

Step 10: go to step 4 and repeat the process until the termination criterion is met. The termination criterion might be that an individual has reached a certain adaption, a maximal number of iterations has been reached, the population has stabilized (i.e., most of its individuals share the same genes, to name a few).

A GA represents an iterative process; each iteration is called a generation. The typical number of iterations for a simple GA can range from 50 to more than 500. Since GAs use a stochastic search method, the fit-

ness of a population can be maintained for a number of generations until the superior individual arises.

III.5.2 Population representation, fitness function and genetic operators

In this section we describe some of the basic elements that are part of the previously described genetic algorithm.

III.5.2.1 Representation

Before applying a GA, first we have to encode the parameters of the problem to optimize. GAs do not directly deal with the parameters, they work with the encodings that represent the parameters. Therefore, the representation of the problems is the first important issue in the design of genetic algorithms, this is to say, how to represent the parameters of the problem.

The various representation schemas can produce a different performance in GAs (Chambers, 1999), (Haupt, Haupt, 2004), (Melanie, 1999). There are two usual representation methods that are used: floating point and bit string. The method of choice is binary string, as most of the genetic operators are adapted for this sort of representation and besides, this representation has a heavier impact on the performance of genetic algorithms. In the binary representation of GAs each parameter to be optimized is encoded by means of a fixed-length binary string, so we need to find an encoding function that assigns an actual parameter value of an integer in the interval $[0, 2^{l-1}]$, where l is the length of the binary string. In order to build this kind of function, by and large, the range of each parameter value is decided on the basis of the previous knowledge of the problem, whose parameters we want to optimize. Based on the range and accuracy desired for the optimal value of each parameter, the required length of the binary string can be calculated. The role of encoding function and its inverse (decoding function) is the coding and decoding of a space of possible solutions for a parameter, in such manner that we can change from the values of the actual parameters to a binary string that can be utilized by GAs.

III.5.2.2 Population

Genetic algorithms work with a population of possible solutions; this way, at the beginning, a GA requires an initial population of individuals. The size of the initial population can be fixed, or depending on the algorithm, this can be adapted. There are three ways of gathering the initial population: at random, deterministic and by using other methods. The two first produce solutions at random. The second initializes the population with specified chromosomes; for example, only the chromosomes of 0's and 1's, and successively (O'Reilly, et al., 2006). Also, knowledge of the problem can be utilized and obtain solutions that meet certain requirements. Finally, the initial population can be initialized with individuals found by means of other optimization techniques.

III.5.2.3 Fitness assessment function

An individual's fitness in genetic algorithms is the value yielded by the fitness assessment function. This assessment function measures the quality of chromosomes to solve a problem. Of course, the fitness of the chromosomes least fit to solve a problem is more severely penalized than the fitness of those fitter.

The fitness assessment function acts as an interface between the genetic algorithm and the optimization problem. In the first place, the chromosome has to be decoded and then assessed by the fitness function that yields a value that indicates the fitness of the chromosomes to solve the problem. The fitness assessment function performs an important role in the genetic algorithm, as it provides information on how well a solution carries out the solution of the problem. This information guides the search for a genetic algorithm, and more accurately, the assessment of results of the fitness function to determine the probability that a possible solution has been selected to produce new solutions in the next generation.

III.5.2.4 Crossover operator

Crossover is a genetic operator that combines two chromosomes (parents) to produce one or two chromosomes (descendants). The idea underlying crossover is that the new chromosome can be better than both parents if it takes the best characteristics of each parent. In the first place,

crossover chooses a point in two parents' chromosomes at random and then interchanges the parts of the chromosomes after such point with a crossover probability defined by the user. As a result, two children solutions are produced. If a pair of chromosomes does not breed, then a cloning of chromosome takes place and the descendants are created as exact copies of each parent (Negnevitsky, 2005).

The most common forms of crossover: one-point, two-point, n-point and uniform crossover are shown in figure III.4

III.5.2.5 Mutation operator

This operator represents a change in the gene (figure III.5). Its function is to promote and guarantee that the search algorithm is not trapped in local optima. Mutation uses mutation probability, p_m, previously defined by the user, which is rather low in nature, and is kept low for GAs; generally, it ranges from 0.001 to 0.01. According to this probability, the bit value changes from 0 to 1 and vice versa. This way, a descendant is produced from a single parent (Negnevitsky, 2005).

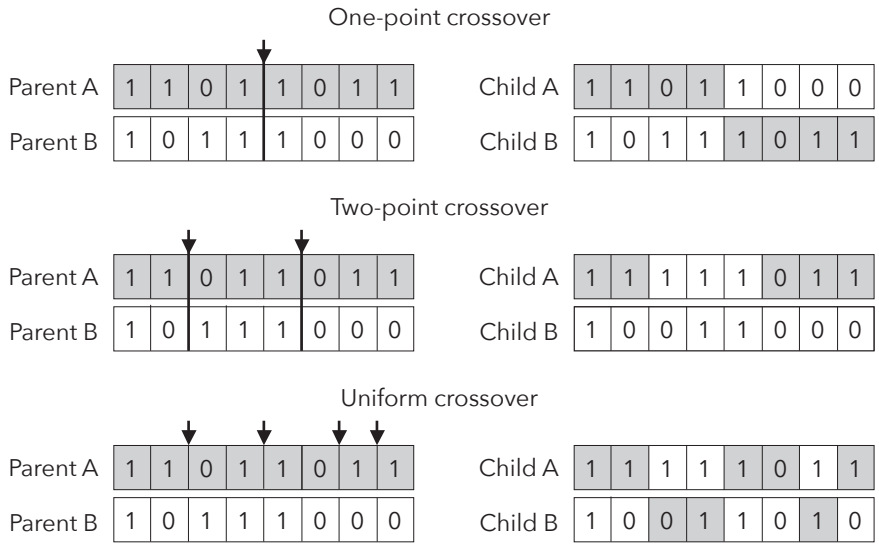


Figure III.4 Crossover operator.

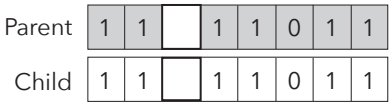


Figure III.5 Mutation operator.

The three next operators compose CHC algorithm (cross-generational elitist selection, heterogeneous recombination (by incest prevention), and cataclysmic mutation) and they convey the idea that crossover shall be the dominant search operator.

III.5.2.6 Elitists selection and incest prevention

After recombination, the only best N individuals have been extracted from the parent population and the descendant population created the new generation. This also implies that duplicated individuals are suppressed from the population. This way of selecting is also known as truncation selection (Eshelman, 1991).

After truncation selection, individuals are paired at random with the new parent population to apply recombination, making N/2 pairs. However, CHC also uses a heterogeneous recombination restriction as an incest prevention method. This is accomplished by pairing those chromosomes that differ from one another in the number of bits, this is to say, a pairing threshold. The initial threshold is established at L/4, where L is string length. If any of the N/2 recombinations could not be applied, i.e., if there is a generation in which none of the new-children population is inserted, then the threshold is reduced in one; this means that chromosomes of the population are very similar.

III.5.2.7 *Half Uniform Crossover* operator

Half uniform crossover (HUX) is an operator in which bits are interchanged at random and independently, but exactly half of the bits that differ between the parents are interchanged, see figure III.6. HUX (Eshelman, 1991) secures that descendants are equidistant from both parents. This works as a mechanism to preserve diversity.

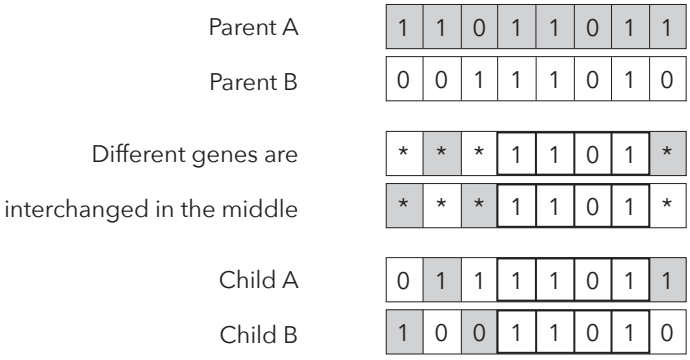
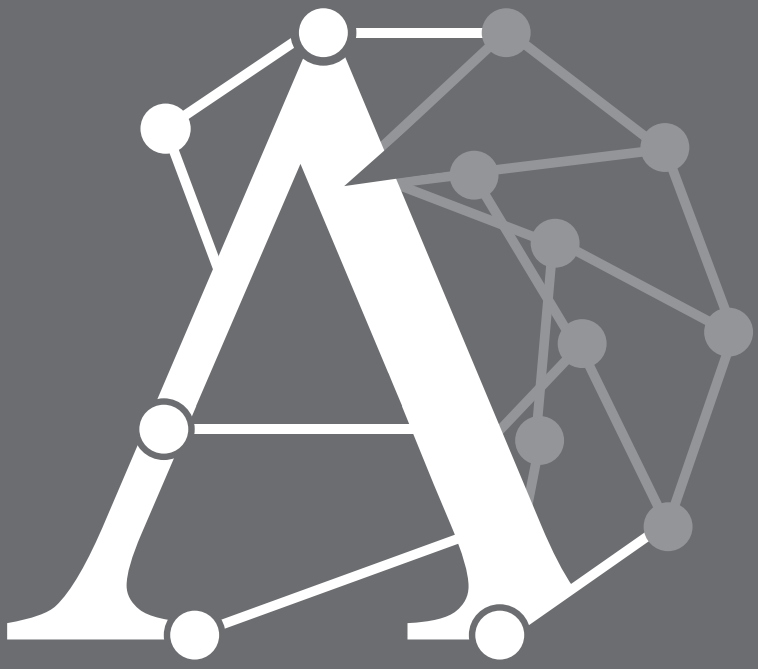


Figure III.6 Representation of crossover HUX operator.

III.5.2.8 Cataclysmic mutation

Mutation is not applied in the phase of normal search of CHC algorithm (Eshelman, 1991). When descendants cannot be inserted in the population of a posterior generation, and the pairing threshold has reached a value of zero, CHC presents new diversity in the population by means of a restarting form. Cataclysmic mutation uses the best individual of the population as a template to initialize population. The new population includes a copy of the best individual; the rest of the population is generated by means of relatively high simple mutation, for example, 35% of the best individual. The new threshold value will be the product of the chromosome length (L) and the mutation percentage (%) used to generate the new population. There are many other ways to refresh population, for instance, in order to rescue the best k individuals and generate the rest randomly, or to rescue the best k individuals and use them as templates to generate the rest of the population, and so successively.



CHAPTER IV.
NEW METHOD TO AUTOMATICALLY GENERATE SINGLE-DOCUMENT SUMMARIES

The objective of this chapter is to present the proposal of a new AGTS method in a single document (section IV.1) and a method to assess the maximal capacity that an AGTS tool can reach (section IV.2). Specifically, in the first section we describe the methods to select and weight terms, to weight and select sentences that will be used in the new AGTS method for a single document independently from vocabulary. It is worth mentioning that the theoretical definitions of the elements comprised in the new proposed method were dealt with in the previous chapter, so in this chapter we only retake the concepts.

IV.1 Method based on MFS and graph weighting for single-document AGTS independently from vocabulary and domain

The presentation of the new method will follow the stages that we have exposed for an AGTS method.

IV.1.1 Term selection

Let us remember; in this stage we must define the term that will be extracted from the text to be able to represent, and with this, quantify the importance of sentences. Considering the terms that can be extracted from the text regardless of language, it is noticed that bag of words is a sub-model of n-grams when n=1. This way, the options would be only between n-grams and MFSs. Since MFSs do not limit their length, as n-grams do, we consider that MFSs perform the analysis by means of all the n-gram models that could be extracted, so only those that are both frequent and maximal will remain.

Our hypothesis is that FSs can express the important and specific ideas of a document. This can be discussed in terms of *tf – idf* (term frequency - inverse document frequency a well-known concept in information retrieval (Baeza-Yates, et al., 1999)): on the one side, the idea expressed by an FS is important for the document if it repeatedly

appears (high term frequency); on the other side, the corresponding idea must be specific for this document. On the contrary there would not be a single word or at least an abbreviation to express it in the entire document (high inverse document frequency).

An *n*-gram can be part or not of another longer *n*-gram. All the n-grams contained in an FS are also FSs. Nevertheless, with the previously stated argumentations it can be inferred that such shorter n-grams might not have any important meaning on their own: for instance, the United States of America is a proper name that represents a state, while States or of America are not. Exceptions such as the United States should not affect too much our reasoning, as they tend to be a synonym of the longest expression and the author of the document would choose a way or another to refer to this country, so these synonyms should not appear too often in the text.

It is important to bear in mind that MFSs represent in a compact manner the set of FSs, since it is possible to reproduce all the set of FSs from MFSs if each MFS is broken down in all its subsequences. This property shall be taken into account, because if a sentence has some subsequences of an MFS it also represents a degree of belonging, it may be too decisive to expect the MFSs to appear as such in the sentence. This is why further in the text we will try a number of sub-term schemas, all of them derived from MFSs in views of finding what would be better.

What we mean by terms is the characteristics that we use in this step. Terms are words, n-grams or MFSs extracted from a document. We also extract terms derived from MFSs, such as words and n-grams. In short, we propose the following variants for term selection:

M: the set of all MFSs, i.e., an n-gram $m \in M$ if it is an MFS with some threshold β (MFSs of two or more words are considered and $\square \geq 2$). In the example in figure IV.1, M = {*is the most beautiful*}. Moreover, we denote with M_2 the set of all MFSs with $\beta = 2$.

B: repetitive bigrams, this is, bigrams with a frequency of at least 2. It is easy to demonstrate that it is the same set as the set of all MFSs bigrams: a bigram $b \in B$ if and only if there exists an MFS $m \in M$ such that $b \subseteq m$. More so, bearing in mind in the last definition M_2 instead of M also yields the same set. In our example, B = {*is the, the most, most beautiful*}.

W: single words (unigrams) of elements from B or, in short, from M. This is to say, a word $w \in W$ if there exists a bigram $b \in B$ such that $w \in b$; it is easy to demonstrate that $w \in W$ if and only if there exists an MFS $m \in M$ such that $w \in m$. Once more, bearing in mind M_2 in

the last definition also yields the same set. In our example, B = {*is, the, most, beautiful*}.

N: all n-grams of MFSs, this is, an n-gram $n \in N$ if there exists an MFS $m \in M$ such that $n \subseteq m$ (including individual words, i.e., 1-grams). Once more, bearing in mind that in the last definition M_2 also yields the same set; this enables the efficient calculation of set N in practice. In our example, N = {*is, the, most, beautiful, is the, the most, most beautiful, is the most, the most beautiful, is the most beautiful*}. Consider $W \subset N$, $M \subset N$.

N \ W, N \ M₂, N \ (W ∪ M₂): as in N but without including 1-grams, the entire MFS or both; here M_2 is the set of MFSs with $\beta = 2$. In our example, $N \setminus (W \cup M_2)$ = {*is the, the most, most beautiful, is the most, the most beautiful*}.

A. Mona Lisa is the most beautiful picture of Leonardo da Vinci
B. Eiffel tower is the most beautiful tower
C. St. Petersburg is the most beautiful city of Russia
D. The most beautiful church is not located in Europe

Figure IV.1 Example of 4 sentences from an arbitrary text.

We give the various definitions of sets B and W in order to show that they naturally derive from the concept of MFS and at the same time they can be efficiently calculated.

IV.1.2 Term weighting

We propose an MFS weighting schema that considers the T_i frequency of MFS. The length of MFS and the frequency of terms derived from MFS. T_i terms can be weighted in various manners and have a weight t_i. This general schema is defined as $p_i(t_j) = X \cdot Y$, where $p_i(t_j)$ - j term weighting in documents i, X and Y can be determined as the MFS frequency, MFS length and the frequency of terms derived from SFM. This weighting schema allows detecting which of the MFS characteristics contributes better to summarize a text. In short, we propose the following term weighting schemas:

f: term frequency in MFSs, this to say, the times the term appears in an MFS. In the example in figure IV.1, f (*is*) =3 , as it appears 3 times in the MFS (*is the most beautiful*).

If the term itself is an MFS, then this is only the frequency of this term in the text (for instance, for M, f is the same as the term weight in

experiment 1, for *W* and *N* it is not). Under certain realistic conditions (MFSs do not cross in the text, words do not repeat in the MFS) *f* is the number of times the term appears in the text as a part of a repetitive bigram. In the example of figure IV.1, *f* (*is*) = 3, as it appears three times in a repetitive bigram "*is the*" (and once more in a non-repetitive context: *church is not*).

L: maximal length of an MFS that contains the term. In the example in figure IV.1, *L*(*is*) = 4, as it is contained in a 4-gram, MFS: *is the most beautiful*.

1: the same for all the terms.

IV.1.3 Sentence weighting

At this stage we propose two options to explore, one simple that consists in adding the relevance of the terms, and another more sophisticated, which consists in weighting based on graphs.

IV.1.3.1 Addition of term relevance

The objective of using this sort of weighting is to learn how much the proposed method improves when considering MFSs as terms together with the various weightings that can be verified. For this stage, the addition of the weights contained in the sentences is calculated. When a sentence S_i has a weight s_j = sum w_{ij}, the contribution of T_i to S_j is w_{ij} = $f_{ij} \cdot t_i$, where *f* is a presence of T_i in D_j, *t* is an importance of T_i. Here *f* is binary.

IV.1.3.2 Graph-based weighting

One of the main contributions of this method is the proposal to use TextRank algorithm to weight sentences. Even if TextRank already proposed it, however in terms of words, in this case we propose to use MFSs as the graph nodes and variants of the previously proposed weightings will be calculated as the edges. Finally, the sentences will be weighted at the stage of sentence weighting using PageRank for a non-directed graph (more details on PageRank in Section III.3).

Nodes. We propose to use MFSs as the graph vertexes.

Edges. The relations that connect MFSs are the term weighting relations, such as the frequency of MFSs in a text, length of MFSs and their presence.

Algorithm. We utilized PageRank, a ranking algorithm based on graphs (the version for this text is TextRank) to find the ranking on the nodes in the graph. Iterate the ranking algorithm until it converges. Order the nodes based on their final score. Use the values associated to each node for ranking/selection decisions.

Algorithm: the main steps are:

1. The graph vertexes will be each of the sentences, represented by terms derived from MFSs, previously proposed in section IV.1.1.
2. Edges between the vertexes of the non-directed graph will be created considering the various weightings proposed in section IV.1.2.
3. Calculate the first weighting on the basis of the edges and then apply PageRank to weight the nodes of the graph to find their pertinence for a summary. Iterate the weighting algorithm until it converges. Order the edges based on their final score. Use the values associated to each edge for weighting/ selection decisions.

In figure IV.2 we show the graph built considering MFSs as terms (term named M in section IV.1.1) of the sentences in figure III.1. As term weighting the variant of $weight(t_k) = f(t_k)^{L(t_k)}$ was used, which means that the MFS frequency powered to the length of such MFS is taken as the relevance of the term t_k. The initial weight of each edge between V_i and vertex V_j was calculated as:

$$\text{Weight Edge } (V_i, V_j) = \prod t_{k=\{t|t\in V_j \cap t\in V_j\}} \text{weight } (t_k) \qquad (4.1)$$

For instance, considering MFSs as terms of the sentences in figure III.1, the graph in figure IV.2 is built; in this graph, the weight of each edge between two sentences can be calculated as: *Weight Edge (Vi,Vj)*, for instance, sentence *B* has the MFSs {*a, cultural heritage, were*} while sentence *D* has {*were, the pharaohs, the pyramids of Egypt*}, so their intersection would be {*the pyramids of Egypt*}. This way, EdgeWeight (B, D) = 2^4=16.

To calculate the initial relevance of each vertex all the edges of such vertex are added as shown in the following formula (4.2):

$$\text{Initial Weight}(V_i) = \sum_{i \neq j} \text{Edge Weight}(V_i, V_j) \quad (4.2)$$

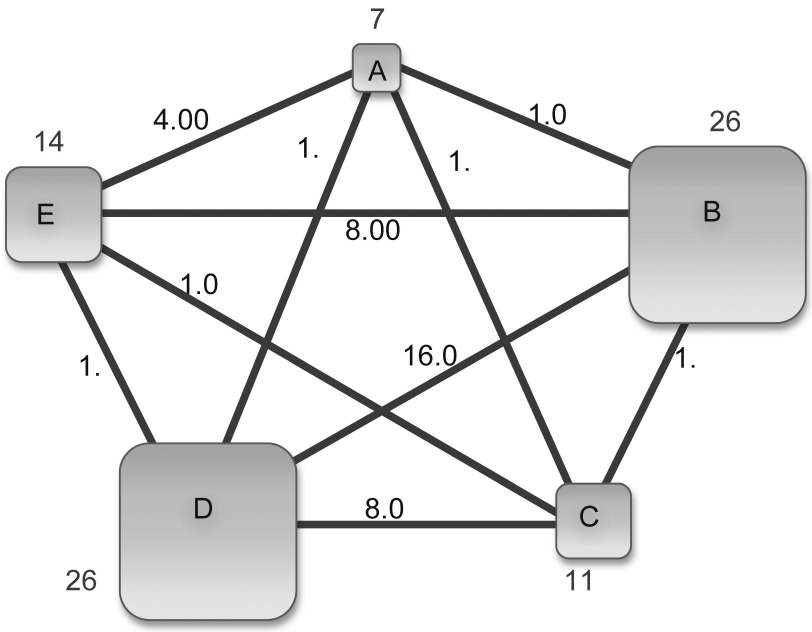


Figure IV.2 Representation of the initial graph using MFSs as terms of the sentences in figure III.1. The size of the node represents the initial importance of the sentence in the document.

Considering the previous example, the vertex *B*=1+8+16+1=26. As it is noticed in figure IV.2, the vertexes can be ordered from the most to the least important as *B, D, E, C, A*. If only one sentence would have to be selected as a summary, there would not be a way to decide on *B* and *D*, as they have the same relevance. If three sentences would have to be selected, they would be *B, D* and *E*.

However, the final relevance of each sentence is obtained until PageRank is applied to the graph, which would be as in figure IV.3, where one notices that the relevance of the vortexes slightly changed, being

clear that sentence *D* is the most important, followed by *B*. This way, the sentences would be ordered, according to their relevance as: *D, B, E, C, A*.

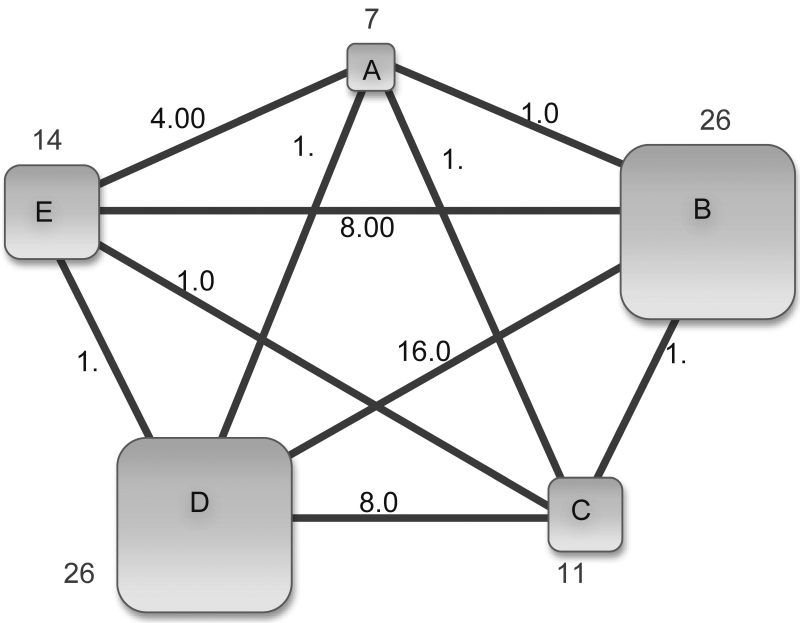


Figure IV.3 Representation of the final graph using MFSs as terms of the sentences in figure III.1. The size of the node represents the importance of the sentence in the document.

In order to measure how much the results reached for the previous example differ from those of the surveyed people Manhattan distance can be used (it was explained in section III.4.1). The distance represents the difference between both criteria, were they identical, it would be zero. In this case, the combination *D, C, B, E, A* yielded 5 that is lower than the combination *C, D, B, E, A* that yielded 6, and which is lower than the result by TextRank in its original version that was 10. This way, it is noticed that the proposal of using MFSs agrees more with the human criterion, at least in this little example. This is to say, seemingly TextRank distances doubly from the proposal, however there are still tests to run with various real documents.

Following the example of the survey, the people were asked to choose two sentences from figure III.1 as a summary. The answers, ordered according to received votes, were: *D*(7), *B*(6), *C*(3), *E*(2) and *A*(2).

Clearly *D* and *B* were the most preferred to be summaries, which also agrees and in the same order with the results of the proposed method. It is also worth mentioning that the proposed method indicated the same sentence the surveyed chose as the worst. As a matter of fact, there is only one vote of difference (sentences *C* and *E*) between the proposed method and the surveyed people's criteria.

IV.1.4 Sentence selection

This procedure completes a summary adding the best weighted sentences or selecting the position of a sentence in a text until the summary reaches the desired number of words.

As in the first option the sentences with a heavier weight are selected. This sort of method is independent from domain and can be applied for a variety of texts. As the second option, the sort of method depends on the position and can only be applied to particular topics. These two options are summarized as follows:

– *best*: sentences with a heavier weight were selected until the desired size is reached (100 words). This method is standard practice.

– *k*best+first: the best k sentences are selected and then the first sentences of the text weighting are selected until the desired size of the summary is reached. This was motivated by *baseline* (mentioned in section V.1) that is very hard to overcome: only the best sentences according to our weighting system can be on top of this *baseline*.

IV.2 New method to calculate *topline* using a genetic algorithm

In this section we present the method proposed to find topline, this is to say, the best combination of candidates to be part of the summary. Likewise, the best result obtained for the collection is called topline. Our method can be applied to find topline not only for the corpus of summary generation (such as DUC-2001 to DUC-2007), but also for other natural language processing tasks. The schema of the proposed method is shown in figure IV.4. In this section, we have the detailed the algorithm of the proposed genetic algorithm.

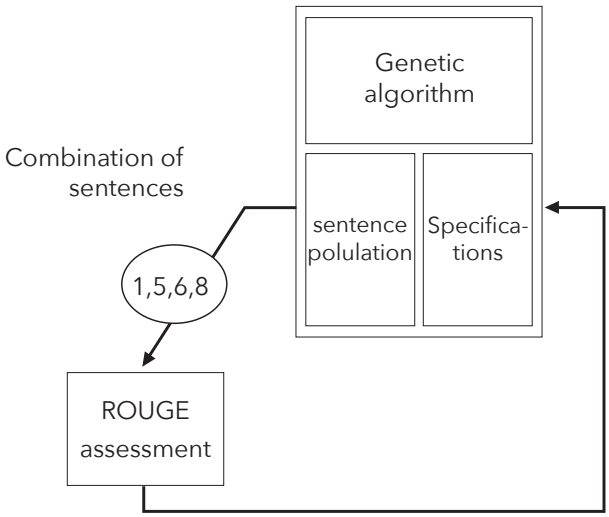


Figure IV.4 Schema of the proposed genetic algorithm.

The genetic algorithm maintains a chromosome population, each of them represents a combination of possible candidates. This genetic algorithm uses data from ROUGE system to assess the fitness of each sentence in the population. This assessment is performed at every time step by means of simulation with each sentence combination and the formation of a fitness function based on ROUGE assessment, which characterizes the desired performance. Using this fitness assessment, the genetic algorithm spreads to the number of sentences of the next generation through of the combination of the genetic operations that are proposed below. The fittest combination of sentences in the population is utilized to compose a summary.

The procedure proposed to estimate the best sentence combination by means of a GA is summarized as follows (see figure IV.5):

1. Determine the number of sentences of the text.
2. Build an initial population.
3. Encode the chromosome in the population.
4. Assess the fitness value of each chromosome.
5. Breed chromosomes according to the fitness value calculated in the previous step.
6. Create children and replace the parents' chromosomes with the children's by means of crossover and mutation.
7. Go to step 3 until the maximal number of iterations is reached.

Representation

To represent the combination of sentences, we use chromosomes of length $N \cdot B$, where N is calculated as the number of sentences in the original text and B the number of bits we use to encode the number of sentences.

Population

The initial population is gathered at random. Its size is fixed and equal to 35 individuals.

Breeding

Once the assessment has been performed, we go on to the breeding stage. We consider to apply the most approximate measure to a balance between diversity and convergence. Such a strategy or algorithm (for instance CHC) implies the use of HUX. This way, the new population is obtained using HUX operator that secures the children are equidistant from the parents. This works as a mechanism to preserve diversity.

Fitness function

We propose the fitness function to measure F-measure for each combination of sentences by means of ROUGE assessment system. And the combination of sentences, which obtain the best score in F-measure, will be the best summary of a text.

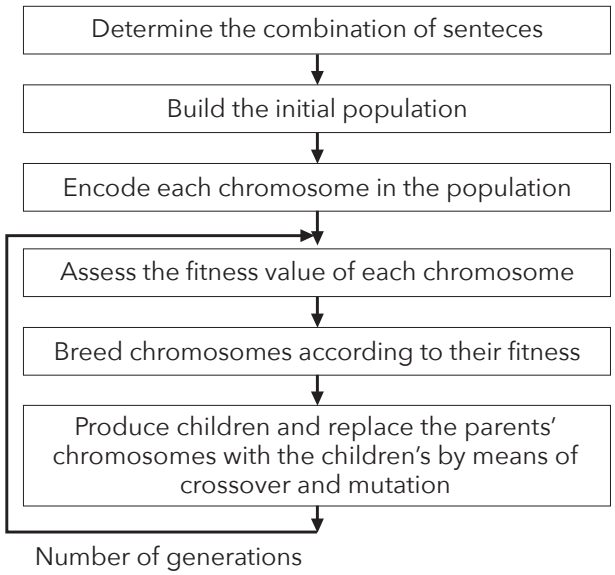
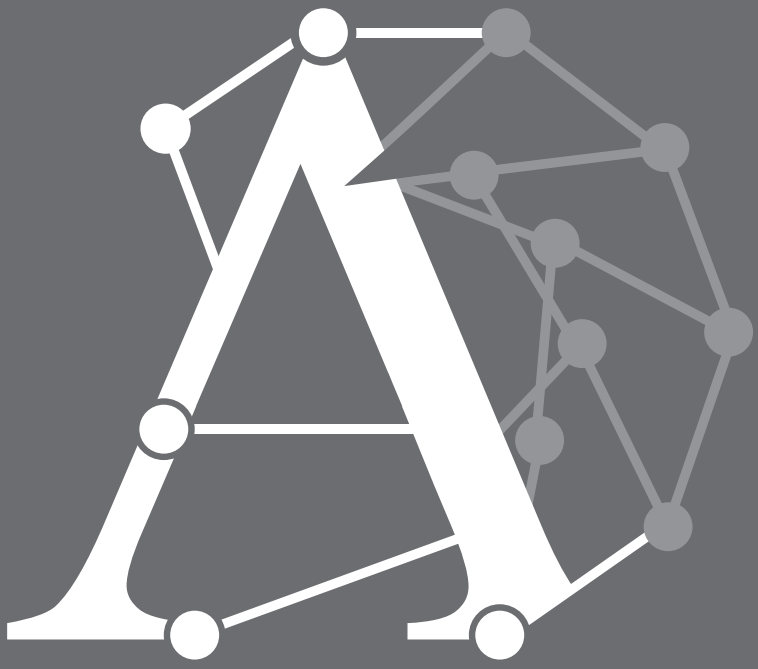


Figure IV.5 Proposed genetic algorithm.



CHAPTER V.
EXPERIMENTAL RESULTS FOR THE AUTOMATIC GENERATION OF SINGLE-DOCUMENT SUMMARIES

In this chapter we experiment with the new method proposed to produce summaries of documents for DUC-2002 corpus; let us bring to mind that it is a standard collection of summaries in English that is utilized to compare results from various text summary generation methods. In this chapter, we try the different schemas to select and weight terms, to weight and select sentences proposed in the previous chapter. For each experiment, we applied the corresponding proposed method and the accomplished percentages were assessed. For each result there is a discussion on the attained percentages. In the second section, we present the results from topline for DUC2002. With the calculation of Baseline:random (the worst quality that can be obtained for AGTS by a method without intelligence) and topline (the best possible quality that a method can obtain); it was possible to recalculate the results of all the methods and tools with the objective to observe how significant the advances that show their results are.

V.1 Elements for experimentation

We undertook a number of experiments to verify the hypotheses formulated in previous chapters.

V.1.1 Algorithm

In each experiment, the standard step sequence was followed:

Term selection: decide which characteristics will be utilized to describe the sentences.

Term weighting: decide what the importance of each calculated characteristic is.

Sentence weighting: decide what the importance of the combined characteristics is regarding the importance of a sentence.

Term selection: decide which sentences are selected for the summary.

The specific adjustments for each step vary in the experiments and are explained in section V.2

V.1.2 Set of texts

The document collection DUC-2002 (DUC, 2014) was utilized; it comprises 567 items of news, in various lengths and dealing with various topics.

Each document in DUC collection is supplied with a number of summaries by two human experts. Although each expert was asked to generate summaries with different lengths, we have only used the 100-word options, so we have two man-made summaries per item in the collection.

V.1.3 Assessment tool

We used ROUGE (Lin, Hovy, 2004) as an assessing tool. This system has the highest correlation with human judgments (Lin, Hovy, 2003). It compares the summaries generated with the program with the human-generated summaries (gold standard). For the comparison, we used the ngram configuration (1, 1) of ROUGE (see section II.3), which had the highest correlation with human judgements, reaching a 95% confidence level.

V.1.4 Baseline

We denoted *Baseline: first* as the selection of the first sentences of the original text until the desired size is reached (DUC, 2014). This configuration produces very good results in the sort of news texts we have been experimenting with. Therefore, we have proposed another baseline (which we believe is a more realistic baseline for other texts different from news items). It is denoted baseline: random, which selects sentences from the original text at random. The presented results are averages of 10 experiments or runs of the experiment (see results in table V.5)

V.2 Experimental methodology

We try various configurations of term selection and weighting, sentence weighting and selection. The following experimental methodology is proposed:

Experiment 1: different options to select terms are tried.

Experiment 2: term selection using MFSs and extracted derived terms.

Experiment 3: term selection using multiword description extracted from a sentence collection (this is to say, for a complete document), term weighting and sentence selection.

Experiment 4: term selection, term weighting and sentence selection using different thresholds.

Experiment 5: term selection, term weighting, and sentence selection with the preprocessing stage.

Experiment 6: term selection, term weighting and sentence selection using graph algorithm.

V.3 Experimental results

According to the aforementioned experimental methodology, we present what the experiment consisted in, the obtained results and the partial conclusions that can be drawn from such results.

V.3.1 Experiment 1

To select the term, we compared MFSs with the most traditional characteristics such as individual words and n-grams. Optionally, stop words were suppressed in the preprocessing stage; in this case our bigrams (or MFSs) might include more words in the original text, as it is explained in chapter III. For term weighting, the frequency of the term was used; for sentence weighting, the addition of the weights of the terms contained in the sentence was used; for sentence selection, the sentences with the heaviest weight were selected until the desired summary size was reached (100 words).

Discussion

As a revision of statistical significance, the data were divided at random in two halves and we ran this experiment (and most of the rest) separately for each group. These experiments verified the qualitative observations reported in this experiment.

As shown in table V.1, MFSs are a promising option for term selection. This motivated our next experiments with term selection schemes derived from them, as well as the options of term weighting for them.

Table V.1 shows the results. Recall or precision measurements can be used for comparison, as the size of all the summaries is the same (100 words). One example from the full text accompanied by summaries produced by a human and the other by the systems is shown in Annex D.

Table V.1 Recall for 100-word summaries for different term selection options

Terms	With stop words	Without stop words
W: B or M words	0.39421	0.41371
B: repetitive bigrams	0.40810	0.42173
M: all MFSs	**0.43066**	**0.44085**

V.3.2 Experiment 2

Inspired by the previous results, there has been more experimentation with MFSs and other options for term selection derived from them. In addition to M, we consider a W option of section III.

The results are shown in table V.2. We carried out the experiments in three stages. From table V.1, we knew that the M term selection schema with suppressed stop words yielded the best results with other fixed parameters (term weighting, sentence weighting and sentence selection). This way, we started from the modification of these parameters in the term selection system; see the upper third of table V.2. The first line of the table represents the best result of table V.1. The best results are highlighted in bold.

In each experiment, we consider the following configuration of the main algorithm:

Preprocessing: optionally, stop words were suppressed at prepossessing stage.

Term selection: each original text is represented separately per sentence. MFSs are extracted from each sentence separately. The resulting multiword descriptions extracted from each sentence are different from the ones extracted from a complete document. Specifically, the representation of a text is different, with the consequence that resulting MFSs are different.

Term weighting: the term frequency in MFSs (*f*); the maximal length of an MFS that contains the (*l*) term; the same weight for all terms (1).

Sentence weighting: the addition of the weights of the terms in the sentence was used.

Sentence selection: to produce the summary, the sentences with the heaviest weight were selected until the desired size was reached (best). In other schema, the best *k* sentences were selected (*kbest*) and then the first sentences were added until the desired text size was reached (*kbest+first*).

Discussion

Later on, we tried other options for term selection, such as *W*, with option 1 for term weighting and the options related to *f*, which showed good performance in the first experiment. Results are shown in the third half of table V.2. W term selection produced a better result than M. Finally, with the best combinations obtained from the two first experiments, we have tried a number of sentence selection variants; see the bottom of table V.2.

It is noticed that any option of the first sentence selection of *kbest+first* surpasses any combination that the standard sentence selection schema utilizes, with lower *k*, it always yields better results, this is, only the slightest correction to *baseline* improves it. The best result was obtained with individual words derived from MFSs, with their weighting by the corresponding MFS frequency

Table V.2 Results of the experiment in which multiword descriptions are extracted from each sentence.

Terms	Term weighting	Sentence selection	Results		
			Recall	Precision	F-measure
M	l × f	best	0.43734	0.45402	0.44519
	1		0.43881	0.45415	0.44600
	l		0.43824	0.45487	0.44606
	f		**0.44034**	**0.45581**	**0.44759**
	l × l		0.42839	0.44633	0.43685
	l ×× f		0.42588	0.44360	0.43423
W	f	best	**0.44483**	**0.45829**	**0.45134**
	1		0.38367	0.40290	0.39291
W	f	1best+first	**0.46523**	**0.48219**	**0.47344**
		2best+first	0.46214	0.47739	0.46952
M	l	1best+first	0.46306	0.48052	0.47150
	f	1best+first	0.46448	0.48185	0.47288
	1	1best+first	0.46423	0.48143	0.47255

V.3.3 Experiment 3

For each experiment in this section, the following configuration of the main algorithm is considered:

Preprocessing: optionally, stop words were suppressed at preprocessing stage.

Term selection: each original text is presented as a sentence collection. MFSs are extracted from a full text. In this experiment, besides M and W from experiment 2, we considered an N option and the generalization of the sets of N, N \ W, N \ M2, N \ (W ∪ M2). N \ W is a subtraction of the set, it is read as the set of N suppressing the elements of W.

Term weighting: the frequency of the term in MFSs (*f*); the maximal length of an MFS that contains the term (*l*); the same weight for all terms (1).

Sentence weighting: the addition of the weights of the terms contained in the sentence was used.

Sentence selection: to produce the summary, the sentences with the heaviest weight were selected until the desired size was reached (*best*). In other schema, the best *k* sentences were selected (*kbest*) and then the first sentences were added until the desired text size was reached (*kbest+first*).

Then we tried other term section options such as *W* and *N*, with option 1 for term weighting and the options related to *f*, which showed good performance in the first experiment. The results are shown in the third half of table V.3. W term selection yielded a slightly better result than M. The results are shown in table V.3. The W term selection yielded a slightly better result than *M*. Results for *N* are the same with f and 1 as weighting. Other combinations based on N did not yield good results; see table V.5 (stop words were excluded, best sentence selection). Finally, we tried different variations of sentence selection; see last third of table V.3.

Table V.3 Results for different options for term detection

Terms	Stop words	Term weighting	Sentence selection	Recall	Precision	F-measure
M	Excluded	f	best	0.44085	0.45564	0.44796
		1		**0.44128**	**0.45609**	**0.44840**
		l		0.43977	0.45587	0.44752
		l^2		0.42995	0.44766	0.43847
		l × f		0.43812	0.45411	0.44581
	Included			0.43353	0.44737	0.44022
W	Included	f	best	0.44582	0.45820	0.45181
	Excluded			**0.44609**	**0.45953**	**0.45259**
		1		0.38364	0.40277	0.39284
		f^2		0.43892	0.45265	0.44556
N		*f* or 1		0.43711	0.45099	0.44383

W	Excluded	f	1best+first	**0.46576**	**0.48278**	**0.47399**
			2best+first	0.46158	0.47682	0.46895
M		1	1best+first	0.46354	0.48072	0.47185
			2best+first	0.46028	0.47567	0.46772
		l	1best+first	0.46381	0.48124	0.47223
			2best+first	0.45790	0.47430	0.46583

Table V.4 Results for variants of set *N* (options: excluded, *best*)

Terms	Term weighting	Recall	Precision	F-measure
N	*f* or 1	0.43711	0.45099	0.44383
	l	0.42911	0.44324	0.43594
N \ W	1	0.42009	0.43693	0.42823
	f	0.41849	0.43532	0.42662
$N \setminus M_2$	1	0.42315	0.43806	0.43035
$N \setminus (W \cup M_2)$		0.41084	0.42759	0.41893

Comparison with experiment 1

It is noticed that results of experiment 2 are better than those of experiment 3. Therefore, we consider it worth comparing the results of experiments 1 and 3:

State of the art: the author of (Mihalcea, Tarau, 2004), (Mihalcea, 2006) provided us with her data, which we assessed in the same conditions as the proposed methods. Specifically, DirectedBackward of TextRank (Mihalcea, Tarau, 2004) algorithm was assessed. We also presented the results of the original *TextRank* algorithm with the implementation of PageRank with *DirectedBackward* of *TextRank* algorithm, but with the processing of additional data to suppress noisy data (Mihalcea, 2006) and TextRank algorithm with a modified version of PageRank (Hassan, Mihalcea, Banea, 2007). See processing details in (Mihalcea, Tarau, 2004), (Mihalcea, 2006), (Hassan, *et al.*, 2007).

Baseline: we utilized *Baseline*: *first* and *Baseline*: random (see section V.1).

Our proposal: we compared these methods with the best results obtained with our proposal of the sentence selection schemas best and 1best+first, as shown in table V.3. In both cases we obtained the best

results with the option W without stop words for term selection and *f* for term weighting.

For a balanced comparison, we separated the methods according to the sort of information they utilize, in addition to the weighting derived from the terms.

- None (the text is considered a bag of sentences, a sentence as a bag of terms, the terms as strings).

- Order of sentences (for instance, the first sentences are managed in a special manner),

- Previous sophisticated processing to obtain the terms.

We believe that in the future the combination of this sort of information can produce better results. The comparison is presented in table V.5.

Table V.5 Comparison of results of experiment 3 with other methods

Additional information utilized	Method	Recall	Precision	F-measure
None	Baseline: *random*	0.37892	0.39816	0.38817
	TextRank: (Mihalcea, Tarau 2004)	**0.45220**	0.43487	0.44320
	Proposed: *W*, *f*, *best*	0.44609	**0.45953**	0.45259
Order of sentences	Baseline: *first*	0.46407	0.48240	0.47294
	Proposed: *W*, *f*, 1*best+first*	**0.46576**	**0.48278**	0.47399
Preprocessing	TextRank: (Mihalcea, 2006)	0.46582	0.48382	0.47450
	TextRank: (Hassan, et al., 2007)	**0.47207**	0.48990	0.48068

We were not able to apply our method with the preprocessing option, as we did not have access to the specific details of the preprocessing used in (Mihalcea, 2006) and (Hassan, et al., 2007) (see Experiment 5 for preprocessing details). However, in the other two categories this method surpassed the others. Possibly with the same type of preprocessing, our method would surpass the others in the last category as well.

Discussion

We notice that the words in repetitive bigrams are good terms, so does it occur with MFSs (we can speculate that MFSs are still better semantical units, but separating them into words gives a more flexible, less disperse

comparison). For term weighting, it was observed that a good weighting system is the number of times the term appears in the text as a part of a repetitive bigram. With these adjustments, we obtained results superior to those of the state-of-the-art methods; most of them show performances below baseline method, which takes into account a special order in the news items sentences, which contain an almost-ready summary in their first sentences. However, our methods can select a sentence better than baseline (even though the second best sentence selected by our methods was worse than the one by baseline). This produces a hybrid method (a sample sentence and then baseline) superior to both baseline and other state-of-the-art methods.

In this experiment we did not apply preprocessing in spite of having shown being beneficial for other methods, so our results are inferior to those of other methods that apply it, however over them when they do not apply preprocessing. This last makes us think that if we apply preprocessing, we will obtain better results than all the existing methods. This will be one of the experiments described below.

On the other side, our experiments show that utterly different options only affect the global result slightly, at least in the collection we utilized for our experiments. This is probably explained by the nature of the texts in this collection (short news items) and the behavior of the assessing tool, ROUGE: the results obtained with *Baseline*: *random* are quite high (almost any method would produce similar results), while what seemed to produce the best results (select the first sentences of the text) yields rather low results, close to *Baseline*: *random*.

V.3.4 Experiment 4

For this experiment, we used the configuration of the algorithm of experiment 3 (see experiment 3). Then, we tried this configuration with $\beta = 2$, 3, 4 (table V.3). We tried the proposed schemas with $\beta = 2$ (see table V.6), $\beta = 3$ (see table V.7) and $\beta = 4$ (see table V.8). The comparison results are shown in tables V.9 - V.11.

Table V.6 Results for experiment 4 with $\beta = 2$

Terms	Stop words	Term weighting	Sentence selection	Recall	Precision	F-measure
M	excluded	l × f	best	0.43731	0.45347	0.44508
		1		**0.43749**	**0.45182**	**0.44438**
		l		0.43731	0.45347	0.44508
		l^2		0.42781	0.44566	0.43640

W	Excluded	f	best	**0.44659**	**0.45968**	**0.45293**
		1		0.38367	0.40290	0.39291
		f^2		0.44114	0.45512	0.44790
W	Excluded	f	1best+first	**0.46536**	**0.48230**	**0.47355**
			2best+first	0.46296	0.47769	0.47009
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46342	0.48069	0.47177
			2best+first	0.45701	0.47320	0.46484

Table V.7 Results for experiment 4 with $\beta = 3$

Terms	Stop words	Term weighting	Sentence selection	Recall	Precision	F-measure
M	Excluded	l × f		0.43470	0.45120	0.44247
		1		**0.43701**	**0.45310**	**0.44459**
		l		0.43470	0.45120	0.44247
		l^2		0.42686	0.44463	0.43525
W	Excluded	f	best	**0.44397**	**0.45773**	**0.45062**
		1		0.38367	0.40290	0.39291
		f^2		0.43797	0.45220	0.44485
W	Excluded	f	1best+first	**0.46622**	**0.48407**	**0.47486**
			2best+first	0.46223	0.47806	0.46989
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46631	0.48392	0.47483
			2best+first	0.46007	0.47638	0.46796

Table V.8 Results for experiment 4 with $\beta = 4$

Terms	Stop words	Term weighting	Sentence selection	Recall	Precision	F-measure
M	Excluded	l × f		0.43013	0.44680	0.43812
		1		**0.43266**	**0.44861**	**0.44025**
		l		0.43013	0.44680	0.43812
		l^2		0.42354	0.44084	0.43183
W	Excluded	f	best	**0.44631**	**0.46505**	**0.45536**
		1		0.38367	0.40290	0.39291
		f^2		0.43712	0.45138	0.44402
W	Excluded	f	1best+first	**0.46788**	**0.48537**	**0.47634**
			2best+first	0.46397	0.47985	0.47165
M		1	1best+first	0.45674	0.47551	0.46582
		l	1best+first	0.46568	0.48373	0.47441
			2best+first	0.45977	0.47604	0.46734

Comparison with experiment 3

In this experiment we obtained better results with the proposed schemas using different thresholds. Here, we compared the best results of the current experiment (see tables V.9 – V.11). We detected that the best configuration for MFSs as selected terms was obtained with the threshold combination (β = 2, 3, 4). We also detected that for the terms derived from MFSs, the best threshold is β = 2. The results of the configuration with the sentence combination β = 4 is the best result obtained.

Table V.9 Results with MFSs terms and different thresholds

Method	Recall	Recall	F-measure
M where β = 2, 3, 4	**0.44128**	**0.45609**	**0.44840**
M where β= 2	0.43749	0.45182	0.44438
M where β = 3	0.43701	0.45310	0.44459
M where β = 4	0.43266	0.44861	0.44025

Table V.10 Results with terms derived from MFSs and different thresholds

Method	Recall	Recall	F-measure
M where β = 2, 3, 4	0.44582	0.45820	0.45181
M where β = 2	**0.44659**	**0.45968**	**0.45293**
M where β = 3	0.44397	0.45773	0.45062
M where β = 4	0.44090	0.45509	0.44776

Table V.11 Results with sentence combination and different thresholds

Method	Recall	precision	F-measure
M where β = 2, 3, 4	0.46576	0.48278	0.47399
M where β = 2	0.46536	0.48230	0.47355
M where β = 3	0.46622	0.48407	0.47486
M where β = 4	**0.46788**	**0.48537**	**0.47634**

Discussion

There are only five better systems (Mihalcea, 2006) than baseline configuration, with slight differences in the results. In the previous experiment,

we obtained better results than baseline. For DUC2002 the results of baseline configuration are very high because most of the texts are news items and in this sort of documents it is common that first lines briefly describe the item. This is to say, some of the first sentences are summaries of a given text. In other sort of texts, baseline configuration will not work. Therefore, it is fair to compare it with the state-of-the-art methods, as random walks (Mihalcea, 2006). The author of this work provided us with the data of summaries that were assessed under the same conditions as the proposed methods. Specifically, we assessed TextRank in its DirectedBackward version (see table V.12, TextRank). Finally, we include the best results of the proposed methods.

Table V.12 Comparison of results of experiments 2 and 3 with other methods

Additional information utilized	Method	Recall	precision	F-measure
None	Baseline: random	0.37892	0.39816	0.38817
	TextRank: (Mihalcea, Tarau, 2004)	**0.45220**	0.43487	0.44320
	Proposed: *Z*, *best*	0.44659	**0.45968**	**0.45293**
Sentence order	Baseline: first	0.46407	0.48240	0.47294
	Proposed: *Z*, 1*best+first*	**0.46788**	**0.48537**	**0.47634**

We tried various combinations of term selection, term weighting, sentence weighting and sentence selection schemas with different thresholds. In the first experiment, we observed that MFSs are good terms and that they help obtain good results that are compared with words and n-grams. In the second experiment, the proposed schemas were tried with various thresholds. We reached the conclusion that words derived from MFSs are the best terms with $\beta = 2$ and MFSs are good terms with $\beta = 2, 3, 4$.

V.3.5 Experiment 5

The results of the experiment are presented in table V.13 where preprocessing is carried out, excluding stop words. The best results are highlighted in bold. We detected that the weighting system of frequency of words derived from MFSs offers the best sentence of the summary, and together

with the sentences obtained with baseline configuration, the best result is obtained. For the first part of this experiment, stop words were excluded.

Table V.13 Results of one configuration of experiment 2 using preprocessing (stop words excluded)

Term selection	Term weighting	Sentence selection	Results		
			Recall	precision	F-measure
M	l × f	best	0.42689	0.43347	0.43005
	1		0.44193	0.44426	0.44298
	l		0.42263	0.42961	0.42599
	f		**0.44678**	**0.44849**	**0.44752**
W	f	best	**0.45504**	**0.45626**	**0.45553**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46416	0.48090	0.47226
		2best+first	0.46033	0.47532	0.46759
M	1	1best+first	**0.46266**	**0.47979**	**0.47094**
	f	1best+first	0.44605	0.44771	0.44676

For the second part of this experiment, we changed the configuration of preprocessing: MFSs are applied *stemming* and stop words were excluded from MFSs. See results in table V.14.

Table V.14 Results of other configuration of experiment 2 using preprocessing (*stemming* and stop words excluded)

Term selection	Term weighting	Sentence selection	Results		
			Recall	Precision	F-measure
M	l × f	best	0.42538	0.43151	0.42831
	1		0.44315	0.44517	0.44405
	l		0.41837	0.42496	0.42153
	f		**0.44538**	**0.44681**	**0.44598**
W	f	best	**0.45576**	**0.45679**	**0.45615**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46413	0.48081	0.47220
		2best+first	0.46259	0.47721	0.46966
M	1	1best+first	**0.46456**	**0.48169**	**0.47285**
	f	1best+first	0.46432	0.48139	0.47258

For the third part of this experiment: MFSs were applied *stemming* and stop words were included. Results are shown in table V.15.

Comparison with experiment 3

We compared with the state-of-the-art methods such as TextRank (Mihalcea, 2006). Specifically, its DirectedBackward version was assessed under the same conditions as the proposed methods (see table V.16, *TextRank*) and the same version of *TextRank* with preprocessing (see table V.16, *TextRank* with preprocessing). And we also compared the results presented in experiment 3 (see table V.16, MFS without preprocessing). Finally, we included the best version of each experiment (see MFS with preprocessing 1, 2, and 3).

Table V.15 Result of one configuration of experiment 2 using preprocessing (*stemming* and stop words included)

Term selection	Term weighting	Sentence selection	Results		
			Recall	Precision	F-measure
M	l × f	best	0.43386	0.43673	0.43494
	1		0.43971	0.44234	0.44067
	l		0.43380	0.43664	0.43487
	f		**0.43867**	**0.44100**	**0.43949**
W	f	best	**0.44609**	**0.44632**	**0.44608**
	1		0.39657	0.39834	0.39733
W	f	1best+first	0.46486	0.48189	0.47310
		2best+first	0.46293	0.47831	0.47037
M	1	1best+first	0.46461	0.48182	0.47293
	f	1best+first	**0.46508**	**0.48233**	**0.47343**

We observe that preprocessing does not positively affect obtaining terms for the extractive summary, at least not in the case of MFSs.

Discussion

We have modified our automatic method for text summarization for a single document based on MFSs as terms, through the inclusion of pre-

processing. We have found however, that preprocessing does not positively affect the summaries obtained with our method. This is good and bad news. Bad because we have not found better conditions and our summaries did not improve. Good because we verified that classic MFSs (sequences of forms of words and without applying stemming or only important words), which are calculated independently from vocabulary, are good conditions for this task, making our method more robust.

Table V.16 Comparison of the preprocessing result with other methods

Method	Recall	Precision	F-measure
TextRank	0.45220	0.43487	0.44320
TextRank with preprocessing	0.46582	0.48382	0.47450
MFS without preprocessing	0.46576	0.48278	0.47399
MFS with preprocessing 1	0.46266	0.47979	0.47094
MFS with preprocessing 2	0.46456	0.48169	0.47285
MFS with preprocessing 3	0.46508	0.48233	0.47343

On the other side, since we have shown that preprocessing almost does not negatively affect the results, then stop words and stemming can be excluded from processing and still obtain almost the same quality of extractive summaries. Excluding stop words significantly reduces the risk of exponential growth of the size of the data structures utilized to extract MFSs and for the application of our method, as well as the number of terms (MFSs or n-grams) to deal with.

V.3.6 Experiment 6

Once we have proved that MFSs as multi-word terms and the combinations of frequency and length weightings represent the relevance of a term, it will be proved how much the results improve by using TextRank graph algorithm to weight the sentences. In each experiment we consider the following configuration of the proposed method:

Term selection: *M, W.*

Term weighting: the frequency of the term in MFS (*f*); the maximal length of the MFS that contains the term (*l*); the same weight for all terms (1).

Sentence weighting: using PageRank.

Sentence selection: the sentences with heavier weights were selected to reach the desired size of the summary.

The results are shown in tables V.17 and V.18. The size of the summaries is 100 words. For comparison we use F-measure. The best results are highlighted in bold.

In table V.17 we use normalization. More specifically, when the weight of sentences is calculated, the sentence weight is divided between the number of words of the sentence.

Table V.17 Results of graph algorithm (normalization was used)

Term selection	Term weighting	Sentence selection	Results		
			Recall	Precision	F-measure
M	f	best	0.48009	0.47757	0.47865
	f^2		0.48056	0.47801	0.47910
	1		0.46668	0.48337	0.47474
	l		0.48025	0.47773	0.47881
	l^2		**0.48058**	**0.47812**	**0.47917**
	f × l		0.48060	0.47810	0.47916
	f ×× l		0.48079	0.47831	0.47937
W	f	best	**0.48659**	**0.48324**	**0.48473**
	1		0.47682	0.47604	0.47626
	f^2		0.48705	0.48235	0.48451
W	f	1best+first	0.47603	0.47518	0.47543
		2best+first	0.47718	0.47621	0.47652
M	l	1best+first	0.47783	0.47699	0.47724
		2best+first	0.48212	0.48088	0.48132
	f	1best+first	0.47797	0.47712	0.47737
		2best+first	**0.48211**	**0.48093**	**0.48134**

We ran our experiments in three phases. From the results of other methods, we knew that M term selection scheme with suppressed stop words produced the best results with other fixed parameters (term weighting, sentence weighting and sentence selection). This way, we started modifying these parameters in the term selection schema; see the upper part of table V.17.

Table V.18 Results of graph algorithm

Term selection	Term weighting	Sentence selection	Results		
			Recall	Precision	F-measure
M	f	best	0.48803	0.48533	0.48626
	f^2		0.48746	0.48482	0.48572
	1		0.47484	0.49180	0.48283
	l		**0.48823**	**0.48577**	**0.48658**
	l^2		0.48741	0.48518	0.48587
	f × l		0.48796	0.48529	0.48620
	f ×× l		0.48716	0.48497	0.48564
W	f	best	**0.48821**	**0.48424**	**0.48604**
	1		0.47529	0.47483	0.47489
	f^2		0.48784	0.48322	0.48534
W	f	1best+first	0.47694	0.47612	0.47635
		2best+first	0.47870	0.47761	0.47798
M	l	1best+first	0.47711	0.47623	0.47650
		2best+first	0.48064	0.47923	0.47976
	f	1best+first	0.47738	0.47649	0.47676
		2best+first	**0.48148**	**0.48016**	**0.48065**

Then we tried other term selection options, such as W, with term weighting 1 and the options related to *f*, which showed the best performance in the first experiment. The results are shown in the third half of table V.17. For term selection, W gave a much better result than M. We discarded reporting N term selection, as the obtained results were not better. Other combinations based on M and W did not produce good results in comparison with other method in which this option gave the best results; see the bottom of table V.17.

Finally, we discarded the normalization of term weighting and we might obtain better results for M and W term selection, in which M is slightly better than W (see table V.18). It is noticed that any sentence selection option kbest+first does not surpass the standard selection schema. The best result was obtained with MFSs by means of weighting the corresponding MFS length.

In figure V.1 we show the comparison of methods and tools of the state of the art and the three best results of the proposed method. As it is noticed, the first schema (SFMs: k-best) improves the results of TextRank, but not of the best commercial tool. The second schema (SFM:

1best+first) manages to improve the commercial tools; however, this last schema, in spite of being independent from language is still more dependent on the news domain, as it combines the first sentences. Finally, the proposal of MFSs with graphs without preprocessing shows a better performance than the previous works, with the advantage of being independent from domain and language.

In the graph in figure V.1, the dotted line represents the calculation of what the worst method, one that selects sentences at random, can obtain as the bottom baseline (baseline: random). This makes us wonder, what would be the topline that any method wants to reach? Thereby, the statement is how to be able to calculate topline. In the following section we present the proposal for a genetic algorithm to calculate topline.

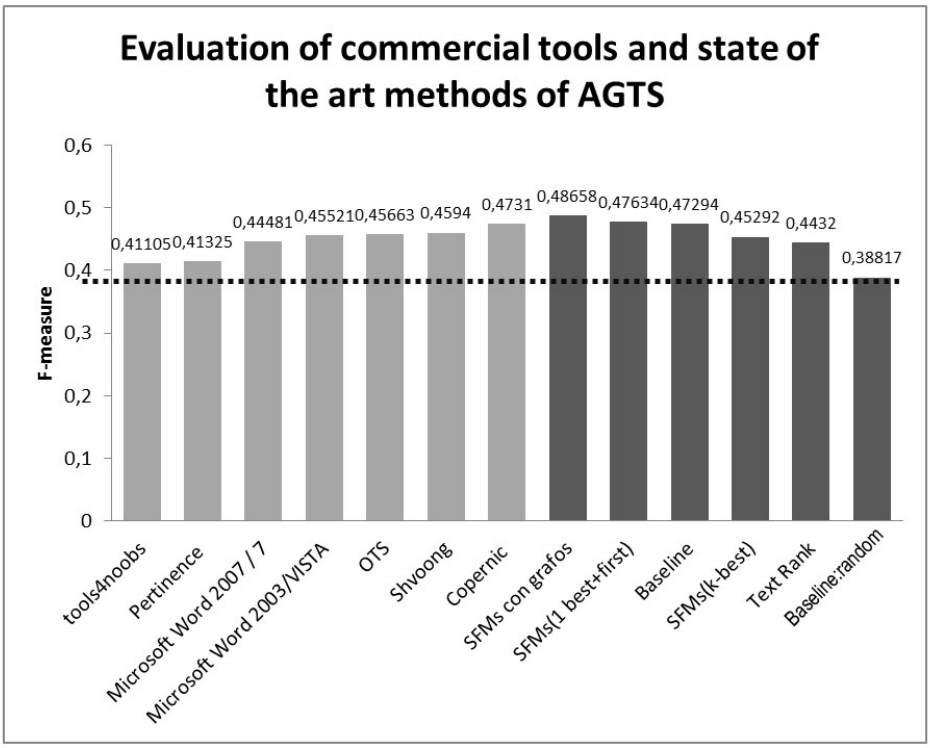


Figure V.1 Comparison of the state-of-the-art methods and tools and with the best results of the method proposed for AGTS.

V.3.7 Topline calculation using genetic algorithms

Considering the 567 items of news in DUC-2002, all the documents in this collection were ordered on the basis of the number of sentences. The first documents of the list are short documents with a small number of sentences. At the end of the list one finds the documents with more words. This list works for assessing by means of ROUGE each of the possible sentence combinations that can be part of a summary. This is made in views of learning which combination produces the best results according to ROUGE with F-measure. Since the number of combinations grows exponentially according to the number of entry sentences, it becomes virtually impossible (owing to the time it takes) to calculate topline for the longest documents. To do so, we propose to use a genetic algorithm that generates a population of possible solutions, which will evolve so that they find a combination that allows optimizing the fitness function, which in this case is ROUGE with F-measure.

In table V.19 we show topline results calculated from the generation of all the possible combinations for the first 400 documents, where the average is 0.6297.

Table V.19 Topline results trying all the sentence combinations

Number of sentences	F-measure
1-49	0.67297
50-99	0.65268
100-149	0.63767
150-199	0.62785
200-249	0.61697
250-299	0.59601
300-400	0.60715
Total	**0.62971**

Later on, we calculated topline for the entire collection, which is shown in table V.20. As noticed in tables V.19 and V.20, the difference between results for the first 400 documents is not much, which validates, to a certain extent, that the results reached by this GA are good. This allows relying on the results that will be attained by the GA for the rest of documents, for which it was not possible to calculate all the combinations.

Table V.20 Topline calculation using the proposed GA

Number of sentences	F-measure
1-49	0.6720
50-99	0.6514
100-149	0.6346
150-199	0.6218
200-249	0.6095
250-299	0.5821
300-349	0.5824
350-399	0.5841
400-449	0.5578
450-499	0.5553
500-549	0.5408
550-568	0.5250
Total	**0.5931**

In table V.21 we show the final results in which the results of tables V.19 and V.20 were combined and from which it was possible to calculate topline for DUC-2002 document collection. Where topline result average was **0.596.**

Table V.21 Topline final results considering all the sentence combinations (0-299) and the proposed genetic algorithm (300-368)

Number of sentences	F-measure
1-49	0.67297
50-99	0.65268
100-149	0.63767
150-199	0.62785
200-249	0.61697
250-299	0.59601
300-349	0.5824
350-399	0.5841
400-449	0.5578
450-499	0.5553
500-549	0.5408
550-568	0.5250
total	**0.5960**

With the results obtained in this research, from topline and baseline: random it is possible to recalculate the results to find out how significant the results reached by each work are, because it seemed according to figure V.1 that a half percentage point is significant. This is to say, according to the results in figure V.1, it looks like too much work only to improve a half percentage point. To recalculate the data, baseline: random was considered 0%, and topline 100%.

As it is noticed in figure V.2, the accomplished advancement is more significant, as it is even noticed that MFSs with graphs is 19.9% better than TextRank. It is also worth mentioning that Copernic, in spite of producing good results is a tool that uses knowledge dependent on language.

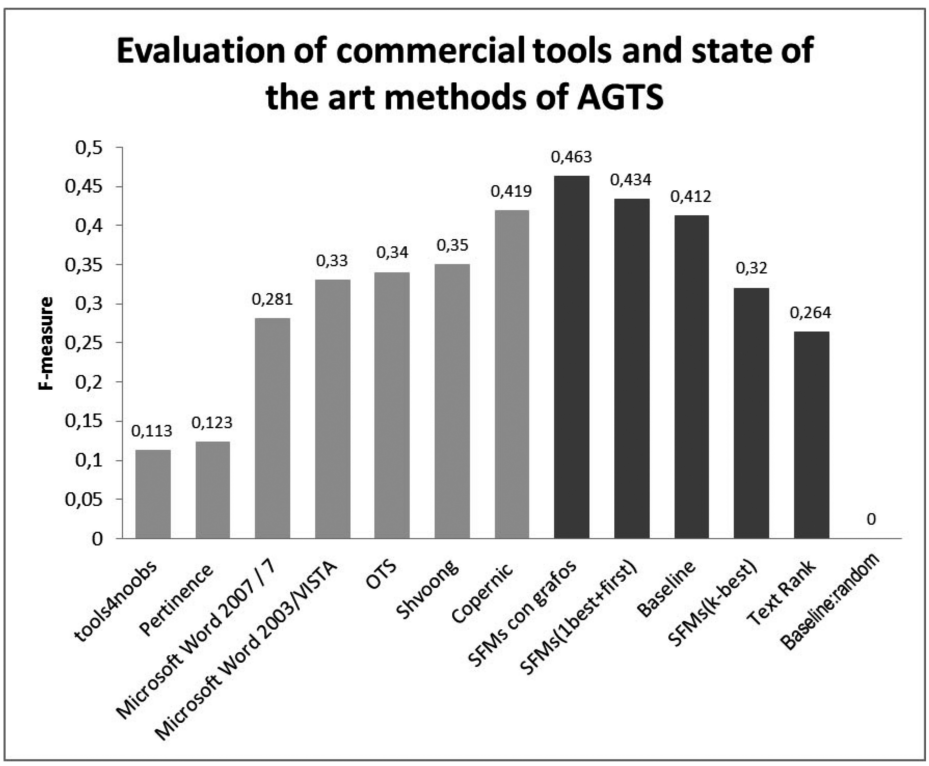


Figure V.2 Significant advancement of the state-of-the-art methods and tools and the best results of the method proposed for AGTS.

CHAPTER VI.
CONCLUSIONS

In this book we presented a panorama of natural language processing focused on the automatic treatment of text in the task of Automatic Generation of Text Summaries (AGTS). Specifically, we deal with AGTS for a single document with the objective of analyzing the possible technologies that, in each of the stages of AGTS, allowed proposing a new method with a dual challenge: on the one side, improve the quality of summaries, and on the other, depend the least possible on vocabulary and domain. The development of these methods for the automatic generation of text summaries enable us to contribute in an efficient manner to the natural language processing area.

The proposed method includes the description of the stages of term selection, term weighting, sentence weighting and sentence selection. For each experiment, the configuration of the proposed methods and the corresponding results were presented. In like manner, the discussion of the experimental results, the comparison between various experiments in this book and the state of the art were explicitly specified.

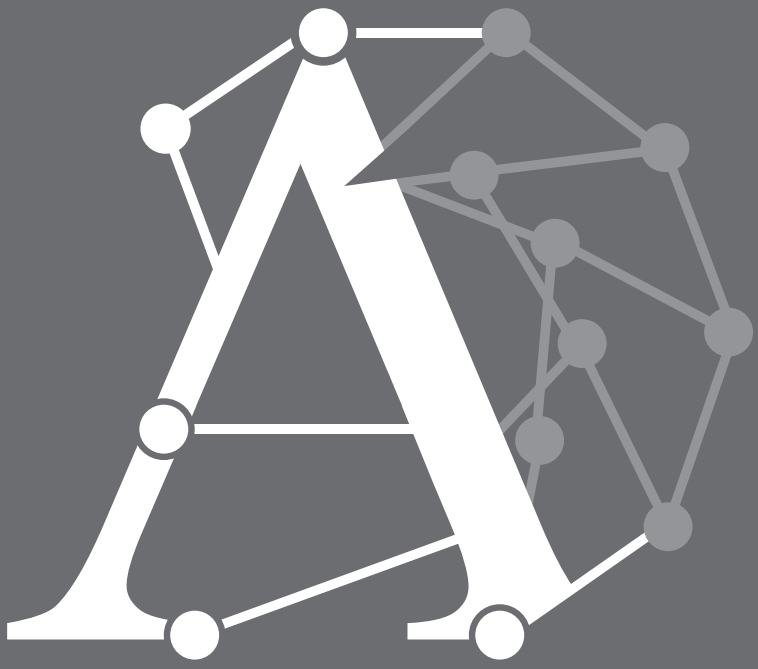
We have accomplished the following contributions. In particular, the following goals were reached:

- Identification of four general stages followed by an AGTS extractive method.
- Assessment of the quality of summaries generated by a large amount of commercial tools and state-of-the-art methods for AGTS for a single document, which establishes a more real state of the art in this research area.

- The calculation of topline in AGTS of extractive kind for a document, which allowed learning, on the one side, that the results of the proposed method significantly improve in relation to other methods and commercial tools. On the other side, this calculation allowed finding out that there is a lot to do in this research area.
- A new method for the automatic generation of extractive text summaries of a single document independent from vocabulary and domain with results superior to those of the state-of-the-art methods and commercial tools.
- For the term selection stage we proposed to utilize maximal frequent sequences as a term independent from language and domain, which enhances their representation because of the multiword descriptions.
- For the term weighting stage several schemas were proposed, but basically we proposed the use of term length as a measure to weight the relevance of the term, from which other combinations can be generated.
- For sentence weighting and selection we proposed to utilize PageRank algorithm to weight the relevance of each sentence according to the sentences the document contains.

The proposed method was tried with DUC-2002. This is a standard collection of summaries in the English language proposed in the conference on text summaries, which makes it easy to compare the results obtained by researchers in the area of text summary generation. However, despite the proposed method, because of the way it works, is independent from language and domain, in the future it will be recommendable to try this method in other domains and languages to find out whether the quality of the obtained summaries is also good.

To answer the research question of this work, we can say that the most important parts of the text can be automatically detected using maximal frequent sequences as multiword descriptions that enhance the extracted terms.



REFERENCES

Aceves-Pérez, R. M., Montes-y-Gómez, M., & Villaseñor-Pineda, L. (2007). "Enhancing cross-language question answering by combining multiple question translations". In *International Conference on Intelligent Text Processing and Computational Linguistics* (vol. 4394, pp. 485-493). Springer Berlin Heidelberg.

Ahonen-Myka, H. (1999). "Finding All Maximal Frequent Sequences in Text". In *Proceedings of the 16th International Conference on Machine Learning* (ICML-99), D. Mladenic and M. Grobelnik (Eds.), (pp. 11–17).

Ahonen-Myka, H. (1999a). "Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery". In *Proceedings of 16th International Joint Conference on Artificial Intelligence* (IJCAI-99), R. Feldman (Ed.), (pp. 1–9).

Ahonen-Myka, H. (1999b). *Knowledge Discovery in Documents by Extracting Frequent Word Sequences. Library Trends on knowledge discovery in bibliographical databases*, J. Qin and M.J. Norton (Eds.), 48(1), (pp. 160–181).

Ahonen-Myka, H. (2002). *Discovery of Frequent Word Sequences in Text, Proceedings of ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, LNCS vol. 2447, ISBN 3-540-44148-4, Springer Verlag, (pp. 180–189).

AMPLN (2014). Asociación Mexicana para el Procesamiento del Lenguaje Natural, consultada el 6 mayo de 2014. http://www.ampln.org/

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co. Inc.

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. In Inderjeet Mani, Mark T. Maybury (Eds.), *Advances in Automatic Text Summarization*, Cambridge/MA, London/England: MIT Press (pp. 111–121).

Barzilay, R. (2003). Information fusion for multidocument summarization: paraphrasing and generation (Doctoral dissertation, Columbia University).

Barzilay, R., & McKeown, K. R. (2005). "Sentence fusion for multidocument news summarization". *Computational Linguistics*, 31(3), (pp. 297-328).

Bolshakov, I., & Gelbukh, A. (2000). A very large database of collocations and semantic links. In *International Conference on Application of Natural Language to Information Systems* (pp. 103-114). Springer Berlin Heidelberg.

Bolshakov, I. A., & Gelbukh, A. (2004a). *Computational linguistics models, resources, applications. Ciencia de la Computación*. IPN-UNAM-FCE, ISBN 970-36-0147-2.

Bolshakov, I. A. (2004b). "Getting one's first million... collocations". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 229-242). Springer Berlin Heidelberg.

Bolshakov, I. A., Galicia-Haro, S. N., & Gelbukh, A. (2005). "Detection and Correction of Malapropisms in Spanish by means of Internet Search". In *International Conference on Text, Speech and Dialogue*, vol. 3658, (pp. 115-122). Springer Berlin Heidelberg.

Bolshakov, I. A., Bolshakova, E. I., Kotlyarov, A. P., & Gelbukh, A. (2008). "Various criteria of collocation cohesion in internet: Comparison of resolving power". In *International Conference on Intelligent Text Processing and Computational Linguistics*, vol. 4919, (pp. 64-72). Springer Berlin Heidelberg.

Brin, S., & Page, L. (2012). "Reprint of: The anatomy of a large-scale hypertextual web search engine". *Computer networks*, 56(18), 3825-3833.

Brunn, M., Chali, Y., & Pinchak, C. J. (2001). Text Summarization Using Lexical Chains. Proc. of Document Understanding Conference 2001. http://duc.nist.gov/pubs.html#2001.

Carbonell, J., & Goldstein, J. (1998). "The use of MMR, diversity-based reranking for reordering documents and producing summaries". *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne, Australia (pp. 335–336).

Carlson, L., Marcu, D., & Okurowski, M. E. (2003). "Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory". In *Jan van Kuppevelt and Ronnie Smith, editors. Current Directions in Discourse and Dialogue*, (pp. 85–112). Springer Netherlands.

Carberry, S., Elzer, S., Green, N., McCoy, K., & Chester, D. (2004). "Extending document summarization to information graphics". In *Proc. of the ACL-04 Workshop: Text Summarization Branches Out* (pp. 3-9).

Carberry, S., Elzer, S., & Demir, S. (2006). "Information graphics: an untapped resource for digital libraries". In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 581-588).

Chuang, W. T., & Yang, J. (2004). Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. Proc. of the ACL-04 Workshop, España.

Chambers, L. D. (Ed.). (1999). *Practical Handbook of Genetic Algorithm Complex Coding System*, CRC Press.

Chali, Y., & Kolla, M. (2004). "Summarization techniques at DUC 2004". In *Proceedings of the document understanding conference* (pp. 105–111). National Institute of Standards in Technology (NIST).

CICLing. Conference on Intelligent Text Processing and Computational Linguistics (2000-2014): www.CICLing.org.

Copernic Inc. Recuperado el 28 de Octubre de 2013, de página web de Copernic. http://www.copernic.com/en/products/summarizer

Corston-Oliver, S., Ringger, E., Gamon, M., & Campbell, R. (2004). "Task-focused summarization of email". In *ACL-04 Workshop: Text Summarization Branches Out* (pp. 43-50).

Cristea D. *et al*. (2005). *Summarization through Discourse Structure*. CICLing 2005, LNCS, vol. 3406, Springer-Verlag, (pp. 621–632).

CV and web-page of Prof. Ph.D. Alexander Gelbukh: www.Gelbukh.com

CV and web-page of Prof. Ph.D. Grigori Sidorov: www.cic.ipn.mx/~sidorov/

Darwin, C. (1956). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. Ed. John Murray.

Daume, H., & Marcu, D. (2004). *Generic Sentence Fusion and Ill-defined Summarization Task*. Proc. of ACL Workshop on Summarization, (pp.96–103).

D'Avanzo E., Elia A., Kuflik T., Vietri S. (2007). *LAKE System at DUC-2007. Proc. of Document Understanding Conference 2007*. http://duc.nist.gov/pubs.html#2007.

Denicia-Carral, C., Montes-y-Gómez, M., Villaseñor-Pineda, L., & Hernández, R. G. (2006). "A Text Mining Approach for Definition Question Answering", *5th International Conference on Natural Language Processing* (FinTal), LNCS, Springer-Verlag, (pp. 76-86)

Dia, Q., & Shan, J. (2006). *A New Web Page Summarization Method*. SIGIR'06. ACM 1-59593-369-7/06/0008.

DUC. Recuperado el 1 de Octubre de 2014, Document understanding conference. http://www-nlpir.nist.gov/projects/duc.

Eshelman, L. J. (1991). "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", In G. Rawlins (Ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, (pp. 265–283).

Evans, D., & McKeown, K. (2005). *Identifying Similarities and Differences Across Arabic and English News*. Proc. of the International Conference on Intelligence Analysis. McLean, VA, 2005.

Farzindar, A., & Lapalme, G. (2004). Legal Text Summarization by Exploration of the Thematic Structure and Argumentative Roles. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, (pp. 27-34).

Ferrández, S., & Ferrández, A. (2007). "The negative effect of machine translation on cross–lingual question answering". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 494-505). Springer Berlin Heidelberg.

Filippova, K., Mieskes, M., Nastase, V., Ponzetto, S. P., & Strube, M. (2007). "Cascaded filtering for topic-driven multi-document summarization". In *Proceedings of the Document Understanding Conference* (Vol. 2007). http://duc.nist.gov/pubs.html#2007.

Futrelle, R. P. (2004). "Handling figures in document summarization". In *Proc. of the ACL-04 Workshop: Text Summarization Branches Out* (pp. 61-65).

Galicia-Haro, S. N., & Gelbukh, A. (2007). *Investigaciones en análisis sintáctico para el español*. Instituto Politécnico Nacional, Dirección de Publicaciones. IPN, ISBN 970-36-0265-7.

García-Hernández, R. A., Martínez-Trinidad, J. F., & Carrasco-Ochoa, J. A. (2004). A Fast Algorithm to Find All the Maximal Frequent Sequences in a Text, 9th Iberoamerican Congress on Pattern Recognition (CIARP), LNCS vol. 3287, (pp. 478–486). Springer Berlin Heidelberg.

García-Hernández, R. A., Martínez-Trinidad, J. F., & Carrasco-Ochoa, J. A. (2006). "A new algorithm for fast discovery of maximal sequential patterns in a document collection". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 514-523). Springer Berlin Heidelberg.

García-Hernández, R. A. (2006). Desarrollo de algoritmos para el descubrimiento de patrones secuenciales maximales, Tesis de doctora-

do, Instituto Nacional de Astrofísica, Óptica y Electrónica, México. 2007.

García-Hernández, R. A., & Ledeneva, Y. (2013). "Single extractive text summarization based on a genetic algorithm". In *Mexican Conference on Pattern Recognition* (pp. 374-383). Springer Berlin Heidelberg.

Gelbukh, A., Sidorov, G., & Han, S. Y. (2003). "Evolutionary approach to natural language word sense disambiguation through global coherence optimization". *WSEAS Transactions on Computers*, 2(1), 257-265.

Gelbukh, A., & Sidorov, G. (2002). "Automatic selection of defining vocabulary in an explanatory dictionary". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 300-303). Springer Berlin Heidelberg.

Gelbukh, A., & Bolshakov, I. A. (2003). "Internet, a true friend of translator". *International Journal of Translation*, ISSN 0970-9819, Vol. 15, No. 2, (pp. 31–50).

Gelbukh, A., Sidorov, G., Han, S. Y., & Hernández-Rubio, E. (2004). "Automatic enrichment of very large dictionary of word combinations on the basis of dependency formalism". In *Mexican International Conference on Artificial Intelligence* (pp. 430-437). Springer Berlin Heidelberg.

Gelbukh, A., & Bolshakov, I. A. (2004). "On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms". In *International Atlantic Web Intelligence Conference* (pp. 105-114). Springer Berlin Heidelberg.

Gelbukh, A., & Sidorov, G. (2006). *Procesamiento automático del español con enfoque en recursos léxicos grandes*, IPN, ISBN 970-36-0264-9.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Addison-wesley.

Hassan, S., Mihalcea, R., & Banea, C. (2007). "Random walk term weighting for improved text classification". *International Journal of Semantic Computing*, 1(04), 421-439.

Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley & Sons.

Hernández-Reyes, E., García-Hernández, R. A., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2006). "Document clustering based on maximal frequent sequences". In *Advances in Natural Language Processing* (pp. 257-267). Springer Berlin Heidelberg.

Holland, J. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.

Hovy, E. (2003). The Oxford handbook of Computational Linguistics, Chapter about Text Summarization, In Mitkov R. (ed.).

Kolla, M., & Chali, Y. (2005). "Experiments in DUC 2005". In *Proceedings of the 2005 Document Understanding Workshop*, Vancouver, Canada.

Kupiec, J., Pedersen, J., & Chen, F. (1995). "A trainable document summarizer". In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 68-73). ACM.

Leavitt, N. (2002). "Data Mining for the Corporate Masses", *IEEE Computer Society Press*, ISSN 0018-9162, vol. 35, Issue 5, (pp. 22–24).

Ledo, Y., Sidorov, G., & Gelbukh, A. (2003). "Tool for computer-aided Spanish word sense disambiguation". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 277-280). Springer Berlin Heidelberg.

Li, J., Sun, L., Kit, C., & Webster, J. (2007). "A query-focused multi-document summarizer based on lexical chains". In Proc. of Document Understanding Conference. http://duc.nist.gov/pubs.html#2007.

Lin, C. Y., & Hovy, E. (1997). "Identifying topics by position". In *Proceedings of the fifth conference on applied natural language processing* (pp. 283-290). Association for Computational Linguistics.

Lin, C. Y., & Hovy, E. (2003). "Automatic evaluation of summaries using n-gram co-occurrence statistics". In *Proceedings of the 2003 Conference of the North American Chapter of the Association for*

Computational Linguistics on Human Language Technology-Volume 1 (pp. 71-78). Association for Computational Linguistics.

Lin, C. Y. (2004). "Rouge: A package for automatic evaluation of summaries". In *Text summarization branches out: Proceedings of the ACL-04 workshop* (Vol. 8).

Lin, C. Y. (2004). "Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?" In *NTCIR*.

Lin, C. Y., & Och, F. J. (2004). "Orange: a method for evaluating automatic evaluation metrics for machine translation". In *Proceedings of the 20th international conference on Computational Linguistics* (p. 501). Association for Computational Linguistics.

Lin, C. Y., & Och, F. J. (2004). "Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics". In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (p. 605). Association for Computational Linguistics.

Liu, D., He, Y., Ji, D., & Yang, H. (2006). "Multi-document summarization based on BE-Vector clustering". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 470-479). Springer Berlin Heidelberg.

Luhn, H. P. (1957). "A statistical approach to mechanized encoding and searching of literary information". *IBM Journal of research and development*, 1(4), (pp. 309-317).

Madnani, N., Zajic, D., Dorr, B., Ayan, N. F., & Lin, J. (2007). "Multiple alternative sentence compressions for automatic text summarization". In *Proceedings of DUC*. http://duc.nist.gov/pubs.html#2007

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing* (Vol. 999). Cambridge: MIT press.

Manning, C. (2007). *An Introduction to Information Retrieval*. Cambridge University Press.

Marcu, D. (2001). "Discourse-based summarization in duc-2001". In *Proceedings of the 2001 Document Understanding Conference* (DUC-2001). http://duc.nist.gov/pubs.html#2001

McKeown, K., Barzilay, R., Chen, J., Elson, D. K., Evans, D. K., Klavans, J., Nenkova, A., Schiffman, B., & Sigelman, S. (2003). "Columbia's Newsblaster: New Features and Future Directions". In *HLT-NAACL* (pp. 15-16).

Melanie, Mitchell. *An Introduction to Genetic Algorithms* (ebook), MIT Press, 1999.

Mihalcea, R., & Tarau, P. (2004). *TextRank: Bringing order into texts*. Association for Computational Linguistics.

Mihalcea, R. (2004a). "Graph-based ranking algorithms for sentence extraction, applied to text summarization". In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (p. 20). Association for Computational Linguistics.

Mihalcea, R. (2006). "Random walks on text structures". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 249-262). Springer Berlin Heidelberg.

Montes-y-Gómez, M., Gelbukh, A., & López-López, A. (2001). "Mining the news: trends, associations, and deviations. Computación y Sistemas", *Revista Iberoamericana de Computación*, Vol. 5, vol. 1, (pp. 14–24).

Montes-y-Gómez, M., Gelbukh, A., & López-López, A. (2002). "Text mining at detail level using conceptual graphs". In *International Conference on Conceptual Structures* (pp. 122-136). Springer Berlin Heidelberg.

Morris, J., & Hirst, G. (1991). "Lexical cohesion computed by thesaural relations as an indicator of the structure of text". *Computational linguistics*, 17(1), (pp. 21-48).

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson Education.

Nenkova, A., & Passonneau, R. J. (2004). "Evaluating Content Selection in Summarization: The Pyramid Method". In *HLT-NAACL*, Vol. 4, (pp. 145-152).

Nenkova, A., Siddharthan, A., & McKeown, K. (2005). "Automatically learning cognitive status for multi-document summarization of newswire". In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 241-248). Association for Computational Linguistics.

Nenkova, A., & Vanderwende, L. (2005). *The impact of frequency on summarization*. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101.

Nenkova, A. (2006). Understanding the process of multi-document summarization: content selection, rewriting and evaluation (Doctoral dissertation, Columbia University).

Neto, J. L., Freitas, A. A., & Kaestner, C. A. (2002). "Automatic text summarization using a machine learning approach". In *Brazilian Symposium on Artificial Intelligence* (pp. 205-215). Springer Berlin Heidelberg.

Otterbacher, J., Radev, D., & Kareem, O. (2006). "News to go: hierarchical text summarization for mobile devices". In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 589-596). ACM.

Open Text Summarizer (OTS). Consultado 1 de Octubre de 2013. http://www.splitbrain.org/services/ots.

O'Reilly, U. M., Yu, T., Riolo, R., & Worzel, B. (Eds.). (2006). *Genetic programming theory and practice II* (Vol. 8). Springer Science & Business Media.

Passonneau, R., Nenkova, A., McKeown, K., & Sigleman, S. (2007). Pyramid evaluation at DUC-05. In Proc. of Document Understanding Conference, http://duc. nist. gov/pubs. html.

Pertinence Summarizer. Consultado 2 de Octubre de 2010, de página principal de pertinence. http://pertinence.net/index_en.html.

Porter, M. F. (1980). "An algorithm for suffix stripping". Program, 14(3), (pp. 130-137).

Radev, D. R., Tam, D., & Erkan, G. (2003). "Single-document and multi-document summary evaluation using Relative Utility". Proc. of the ACM International Conference on Information and Knowledge Management (CIKM-03).

Reeve, L. H., & Han, H. (2007). "A term frequency distribution approach for the duc-2007 update task". In *Proc. of Document Understanding Conference*. http://duc.nist.gov/pubs.html#2007.

Salton, G., Wong, A., & Yang, C. S. (1975). "A vector space model for automatic indexing". *Communications of the ACM*, 18(11), (pp. 613-620).

Salton, G., & Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval". *Information processing & management*, 24(5), (pp. 513-523).

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of Reading*: Addison-Wesley.

Seki, Y. (2002). Sentence Extraction by tf/idf and position weighting from Newspaper Articles. Proc. of the Third NTCIR Workshop.

Shrestha, L., & McKeown, K. (2004). "Detection of question-answer pairs in email conversations". In *Proceedings of the 20th international conference on Computational Linguistics* (p. 889). Association for Computational Linguistics.

Silber, H. G., & McCoy, K. F. (2002). "Efficiently computed lexical chains as an intermediate representation for automatic text summarization". *Computational Linguistics*, 28(4), 487-496.

Song, Y. I., Han, K. S., & Rim, H. C. (2004). "A term weighting method based on lexical chain for automatic summarization". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 636-639). Springer Berlin Heidelberg.

Soricut, R., & Marcu, D. (2003). "Sentence level discourse parsing using syntactic and lexical information". In *Proceedings of the 2003*

Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 149-156). Association for Computational Linguistics.

Svhoong Summarizer. Consultado 2 de Octubre de 2013, de Página principal de Svhoong. http://es.shvoong.com/summarizer

Teufel, S., & Van Halteren, H. (2004a). "Agreement in Human Factoid Annotation for Summarization Evaluation". In *LREC*.

Teufel, S., & Van Halteren, H. (2004b). "Evaluating Information Content by Factoid Analysis: Human annotation and stability". In *EMNLP* (pp. 419-426).

Tools4noobs Summarizer. Consultado el 2 de Octubre de 2013, de página principal de tools4noobs. http://www.tools4noobs.com/summarize/.

Vandeghinste, V., & Pan, Y. (2004). "Sentence compression for automated subtitling: A hybrid approach". In *Proceedings of the ACL workshop on Text Summarization* (pp. 89-95).

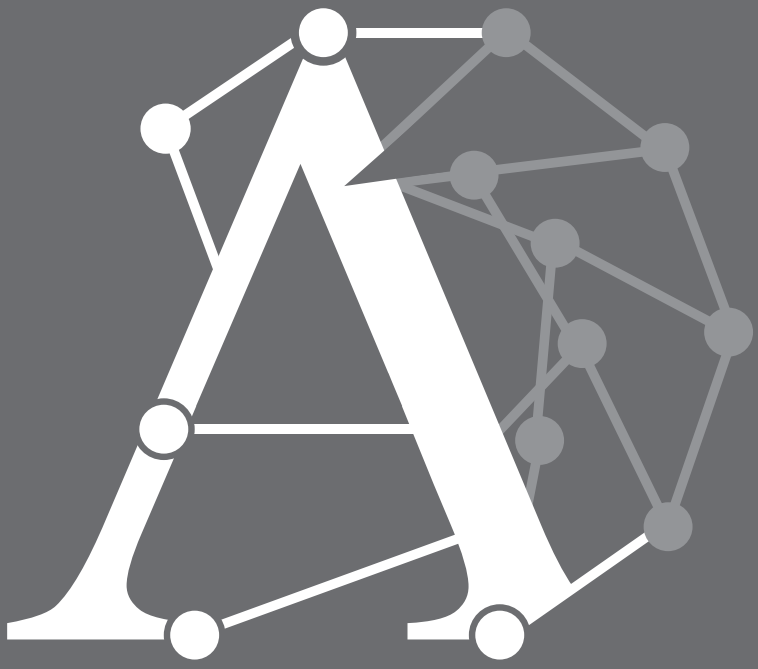
Verma, R., Chen, P., & Lu, W. (2007). "A semantic free-text summarization system using ontology knowledge". In *Proc. of Document Understanding Conference*. http://duc.nist.gov/pubs.html#2007.

Villatoro-Tello, E., Villaseñor-Pineda, L., & Montes-y-Gómez, M. (2006). "Using Word Sequences for Text Summarization". In *International Conference on Text, Speech and Dialogue* (pp. 293-300). Springer Berlin Heidelberg.

Wan, S., & McKeown, K. (2004). "Generating overview summaries of ongoing email thread discussions". In *Proceedings of the 20th international conference on Computational Linguistics* (p. 549). Association for Computational Linguistics.

Xu, W., Li, W., Wu, M., Li, W., & Yuan, C. (2006). "Deriving event relevance from the ontology constructed with formal concept analysis". In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 480-489). Springer Berlin Heidelberg.

Zhou, Q., Sun, L., & Nie, J. Y. (2005). IS_SUM: A multi-document summarizer based on document index graphic and lexical chains. Proc. of Document Understanding Conference 2005. http://duc.nist.gov/pubs.html#2005.



ANNEXES

Annex A. List of stop words

a	doesn't	is	other	there
about	done	isn't	our	theirs
after	due	it	out	them
again	during	its	over	then
all	each	it's	overall	there's
almost	either	itself	per	these
also	enough	just	perhaps	they
although	especially	kg	possible	this
always	etc.	km	previously	those
am	even	largely	quite	through
among	ever	like	rather	thus
an	first	made	really	to
and	followed	mainly	regarding	under
another	following	make	resulted	until
any	for	may	resulting	up
approximately	found	max	same	upon
are	from	me	seem	use
as	further	might	seen	used
at	give	more	several	using
be	given	most	she	various
because	giving	mostly	should	very
been	had	must	show	was
before	hardly	my	showed	we
being	has	myself	shown	were
between	have	nearly	shows	what
both	having	neither	significant	when
but	here	no	significantly	whereas
by	he	nor	since	which
can	he's	not	so	who
can't	her	now	some	while
could	his	obtain	somehow	with
couldn't	how	obtained	such	within
did	however	of	suggest	without
didn't	if	often	than	would
do	I'm	on	that	you
don't	in	only	the	
does	into	or	their	

Annex B. Examples of obtained results

Detailed results for the three best results from table V.3 (see experiment 1):

First result: M, 1, *best*

1 ROUGE-1 Average_R: 0.44128 (95%-conf.int. 0.43352 - 0.44889)
1 ROUGE-1 Average_P: 0.45609 (95%-conf.int. 0.44790 - 0.46415)
1 ROUGE-1 Average_F: 0.44840 (95%-conf.int. 0.44047 - 0.45615)

1 ROUGE-2 Average_R: 0.18676 (95%-conf.int. 0.17845 - 0.19498)
1 ROUGE-2 Average_P: 0.19341 (95%-conf.int. 0.18455 - 0.20230)
1 ROUGE-2 Average_F: 0.18994 (95%-conf.int. 0.18135 - 0.19849)

1 ROUGE-SU4 Average_R: 0.20883 (95%-conf.int.-0.20138 - 0.21582)
1 ROUGE-SU4 Average_P: 0.21618 (95%-conf.int. 0.20873 - 0.22331)
1 ROUGE-SU4 Average_F: 0.21235 (95%-conf.int. 0.20483 - 0.21947)

Second result: W, f, best

1 ROUGE-1 Average_R: 0.44609 (95%-conf.int. 0.43850 - 0.45372)
1 ROUGE-1 Average_P: 0.45953 (95%-conf.int. 0.45160 - 0.46749)
1 ROUGE-1 Average_F: 0.45259 (95%-conf.int. 0.44479 - 0.46048)

1 ROUGE-2 Average_R: 0.19451 (95%-conf.int. 0.18664 - 0.20256)
1 ROUGE-2 Average_P: 0.20048 (95%-conf.int. 0.19229 - 0.20892)
1 ROUGE-2 Average_F: 0.19740 (95%-conf.int. 0.18936 - 0.20566)

1 ROUGE-SU4 Average_R: 0.21420 (95%-conf.int. 0.20755 - 0.22133)
1 ROUGE-SU4 Average_P: 0.22085 (95%-conf.int. 0.21387 - 0.22813)
1 ROUGE-SU4 Average_F: 0.21742 (95%-conf.int. 0.21061 - 0.22462)

Third result: W, f, 1*best+first*

1 ROUGE-1 Average_R: 0.46576 (95%-conf.int. 0.45877 - 0.47292)
1 ROUGE-1 Average_P: 0.48278 (95%-conf.int. 0.47547 - 0.49004)
1 ROUGE-1 Average_F: 0.47399 (95%-conf.int. 0.46693 - 0.48132)

1 ROUGE-2 Average_R: 0.21690 (95%-conf.int. 0.20915 - 0.22497)
1 ROUGE-2 Average_P: 0.22495 (95%-conf.int. 0.21659 - 0.23345)
1 ROUGE-2 Average_F: 0.22080 (95%-conf.int. 0.21278 - 0.22909)

1 ROUGE-SU4 Average_R: 0.23330 (95%-conf.int. 0.22668 - 0.24045)
1 ROUGE-SU4 Average_P: 0.24207 (95%-conf.int. 0.23508 - 0.24941)
1 ROUGE-SU4 Average_F: 0.23754 (95%-conf.int. 0.23075 - 0.24472)

Annex C. Examples of Maximal Frequent Sequences

Flights were cancelled	Personal property
Sunday night	The Yucatan peninsula
Prensa Latina	Tropical storm
Civil defence	Low pressure
Hurricane Gilbert	San Francisco area
The Dominican Republic	Have to pay
The south coast	Insurance companies
The Cayman Islands	Long term
Cancun and Cozumel	State farm
In Mexico City	The big mac
Quintana Roo state	A special session
Over the water	To deal with
Year after year	Department of transportation
The white house	Gasoline tax increase
Structural damage	Might collapse in an earthquake
The California earthquake	The royal marine's music school
Bush and his aides	Military installations
The insurance industry	Private security
Exposure to catastrophes	Irish republican army
On an inflation adjusted basis	Opening day record for
Roamed the streets of Cancun	Restaurant in Moscow
Leader of the Conservative Party	The Soviet Union
The national weather service said	Moscow McDonalds
Above and beyond the usual guest	Walmart discount city
Have earthquake insurance	Vote for Major
Caused coastal flooding	Major was elected
Was elected to parliament	Associated Press
Earthquake insurance	Most important retailer of his generation
Plenty of experience	British Prime Minister John Major
For the purpose of	Vice president of the United States
Whenever I needed him	

Annex D. Examples of automatically generated summaries

In this annex we show two items of news from DUC-2002 collection, each with the summaries produced by expert humans and with the summary automatically generated by experiment 1.

Original text A

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo. Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast. There were no reports of casualties. San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night. On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. Residents returned home, happy to find little damage from 80 mph winds and sheets of rain. Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane. The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Model summary produced by expert 1

Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph. The Dominican Republic's Civil Defence alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Model summary produced by expert 2

Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas. Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

Summary automatically generated by experiment 1

The national weather service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Hurricane Gilbert swept toward the Dominican Republic Sunday, and the civil defence alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. "There is no need for alarm," civil defence director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement.

Original text B

The Irish Republican Army claimed responsibility for a huge explosion Friday that reduced a three-story military barracks on the southeast coast of England to rubble, killing 10 people and injuring 22, eight seriously. It would be one of the outlawed IRA's deadliest attacks on the main British island. Nine marine musicians and one civilian died in the blast, which also damaged dozens of nearby homes and could be heard two miles away. The musicians were between the ages of 16 and 20 as are most of the recruits in the school. A police spokesman said forensic experts are still trying to determine with certainty that the explosion was the result of a bomb. But he said the characteristics of the blast and a statement claiming responsibility appeared to confirm that it was the work of the IRA. Security sources said they believe that at least two IRA "active service" units, each composed of four or five members, are operating in Britain and continental Europe. One member, known as the "Jackal" after the assassin in the Frederick Forsyth novel "The Day of the Jackal," has been eluding the authorities for two years. He has been identified as Patrick Sheehy and has been linked to the IRA's last successful mainland bombing attack -- on an army barracks at Mill Hill in August, 1988. One soldier was killed in that incident. Sheehy and another wanted Irishman, John Conaghty, were linked to an IRA bomb factory in North London that the police stumbled upon last December while in pursuit of a car thief. A search turned up automatic and semiautomatic weapons, ammunition, 150 pounds of Semtex high explosive and a "hit list" of 100 British political figures and other officials headed by Prime Minister Margaret Thatcher. Friday's explosion occurred about 8:30 a.m. in a lounge at the Royal Marines School of Music near Deal, on the English Channel in the county of Kent. At the school are about 250 recruits who receive military and musical training before joining Royal Marines bands. The roof of the three-story barracks collapsed, trapping victims beneath the rubble. Firefighters used thermal cameras and dogs to search the debris for victims and survivors. Heavy lifting gear was brought to the scene from a nearby site where a tunnel is being built beneath the English Channel. Rescuers shouted for quiet as they used high-technology listening equipment in an effort to trace the sound of faint heartbeats. "I looked up from the sink and I just saw the whole building explode," Heather Hackett, a 26-year-old Deal housewife, told the British Press Assn. She said she told her children to run for cover, but as they did, her kitchen window shattered. "The whole window was blown across the kitchen," Hackett recalled. Her 2-year-old son, Joshua, was hit by a shard that embedded itself in his back but caused no serious injury. "I just screamed and ran out of the room," she said. "The bang was

so loud I thought the whole house was coming in." 'Appalling Outrage' Defence Secretary Tom King visited the scene and called the bombing "an appalling outrage committed against unarmed bandsmen -- people who worked for charity, who have given great enjoyment to millions right across the country, right across the world." The real evil of these murders is that the people who commit them, the 'godfathers' who send them to commit them, know that they will actually achieve nothing. Terrorism is not going to win. We shall find the people responsible for this outrage sooner or later, as we have already found some of those responsible for the earlier outrages, and they will be brought to justice." The authorities have been on high alert, expecting IRA attacks in connection with last month's 20th anniversary of the introduction of British troops into Northern Ireland. The republican underground organization opposes British rule in the predominantly Protestant province and is fighting to join the mainly Roman Catholic south in a united, independent Ireland. Visit to Ulster But in a statement telephoned to a Dublin news agency, Ireland International, Friday's attack was linked to Thatcher's visit last week to units of the controversial Ulster Defence Regiment in Northern Ireland. The locally recruited, overwhelmingly Protestant Ulster Defense Regiment has come under fire in connection with an investigation into the leak of secret government lists of suspected IRA members to Protestant assassination squads. It is widely hated by the Catholic minority in the province, and the Irish government in Dublin has urged Britain to disband the force. "Mrs. Thatcher visited Ireland with a message of war at a time when we want peace," the statement claiming responsibility for the Deal attack said. "Now in turn we have visited the Royal Marines in Kent. But we still want peace, and we want the British government to leave our country." The statement was signed "P. O'Neill, Irish Republican Publicity Bureau," a signature that has appeared on earlier IRA bombing claims. Friday's attack was the worst on the mainland since the virtually simultaneous bombings of July, 1982, directed at ceremonial military units in London's Hyde Park and Regent's Park. Eleven bandsmen and mounted guards were killed in those incidents. Eight persons were killed by IRA car bombs outside Harrods department store here in December, 1983, and 21 were killed and 162 injured in two Birmingham public house bombings in the fall of 1974. An attempted barracks bombing was averted last February when a sentry came upon two intruders who had managed to get inside a military camp in Shropshire. There has been a series of bomb and automatic rifle attacks this year on British soldiers and their families stationed in West Germany. Earlier this month an IRA gunman shot to death an army wife, Heidi Hazell, 25, in her car near her home at Dortmund.

Model summary produced by expert 1

A huge explosion yesterday in the lounge of the Royal Marines School of Music killed ten and injured 22, eight seriously. The School is located in Deal on the English Channel. Eyewitness accounts of neighbors attest to the strength of the blast. Investigators said that it was probably a bomb blast, and the IRA has claimed responsibility. The British think that at least two IRA "active service" units, each with four or five members, operate in Britain and continental Europe. Increased IRA activity had been anticipated because last month marked the 20th anniversary of British troops entering Northern Ireland.

Model summary produced by expert 2

In what they said was a response to Prime Minister Thatcher's "declaration of war" in a speech to the Ulster Defense Force, the Irish Republican Army claimed responsibility for an explosion which leveled a three-story barracks in Deal, killing 10 and injuring 22. The barracks, which belonged to the Royal Marines Music School, was the latest in a series of IRA bombings of military facilities. Security forces believe at least two IRA "active service units" are operating in Britain and Europe. Two members of these groups, Patrick Sheehy, known as the Jackal, and John Conaghty are being sought in connection with earlier attacks.

Summary automatically generated by experiment 1

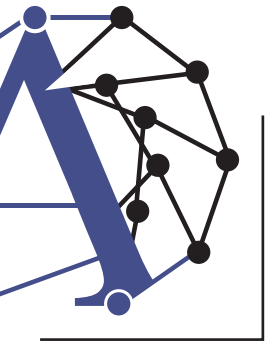
The Irish republican army claimed responsibility for a huge explosion Friday that reduced a three-story military barracks on the southeast coast of England to rubble, killing 10 people and injuring 22, eight seriously. It would be one of the outlawed IRA's deadliest attacks on the main British island. Nine marine musicians and one civilian died in the blast, which also damaged dozens of nearby homes and could be heard two miles away. The musicians were between the ages of 16 and 20 as are most of the recruits in the school. A police spokesman said forensic experts are still trying to determine with certainty that the explosion was the result of a bomb.

Generación automática de resúmenes
Retos, propuestas y experimentos

Automatic Generation of Text Summaries
Challenges, proposals and experiments

de Yulia Nikolaevna Ledeneva y René Arnulfo García Hernández, fue impreso en los talleres de Editorial CIGOME, S.A. de C.V., Vialidad Alfredo del Mazo núm. 1524, ex. Hacienda La Magdalena C.P. 50010, Toluca, México. Su edición consta de 1 000 ejemplares. La edición estuvo a cargo de la Dirección de Difusión y Promoción de la Investigación y los Estudios Avanzados.

Coordinación editorial: Patricia Vega Villavicencio
Corrección de estilo (versión en español): Tomás Fuentes Estrada
Diseño de portada e interiores: Juan Manuel García Guerrero y Cristina Mireles Arriaga



El aumento exponencial de información electrónica ha provocado la necesidad de comprender de manera rápida el contenido esencial de grandes volúmenes de información textual. Normalmente, los propios autores redactan un resumen de su obra, sin embargo son muy pocos los documentos que cuenta con uno. De hecho se estima que el ochenta por ciento de la información electrónica de una empresa se encuentra en forma de texto y el otro veinte por ciento en forma de bases de datos. Este porcentaje aumenta cuando se habla de Internet, puesto que la mayoría de la información se encuentra en forma textual, es decir, en forma de lenguaje natural.

Este libro presenta un método computacional novedoso a nivel internacional, el cual fue desarrollado para generar de manera automática resúmenes de texto independientemente del lenguaje y dominio al que pueda ser aplicable. Los resúmenes generados por el método presentado son los más parecidos al humano en comparación a los resúmenes que actualmente pueden generar otros métodos y herramientas de software del estado del arte.