INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
LABORATORIO DE LENGUAJE NATURAL Y PROCESAMIENTO DE TEXTO

# Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization

# TESIS

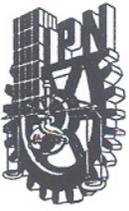QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA COMPUTACION

PRESENTA

## M. en C. Yulia Nikolaevna Ledeneva

DIRECTOR DE TESIS:
## Dr. Alexander Gelbukh

MEXICO, D.F.                                        Noviembre de 2008

# INSTITUTO POLITÉCNICO NACIONAL
## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

*ACTA DE REVISIÓN DE TESIS*

En la Ciudad de __México, D. F.__ siendo las __18:00__ horas del día __05__ del mes de __Septiembre__ de __2008__ se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

### Centro de Investigación en Computación

para examinar la tesis de grado titulada:

### "AUTOMATIC LANGUAGE-INDEPENDENT DETECTION OF MULTIWORD DESCRIPTIONS FOR TEXT SUMMARIZATION"

Presentada por la alumna:

| LEDENEVA | | YULIA NIKOLAEVNA |
|----------|---------|------------------|
| Apellido paterno | Materno | nombre(s) |

Con registro: | B | 0 | 6 | 1 | 5 | 4 | 4 |

aspirante al grado de: *DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN*

Después de intercambiar opiniones los miembros de la Comisión manifestaron *SU APROBACIÓN DE LA TESIS*, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISIÓN REVISORA
Presidente

Dr. Grigori Sidorov

Secretario

Dr. Francisco Hiram Calvo Castro

Primer vocal
(Director de Tesis)

Dr. Alexandre Felixovich Guelboukh Kahn

Segundo vocal

Dr. Luis Villaseñor Pineda

Tercer vocal

Dr. Yoel Ledo Mezquita

EL PRESIDENTE DEL COLEGIO

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION

# INSTITUTO POLITÉCNICO NACIONAL
## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

### CARTA CESION DE DERECHOS

En la Ciudad de México D.F. el día 16 del mes octubre del año 2008, la que suscribe Yulia Nikolaevna Ledeneva alumna del Programa de Doctorado en Ciencias de la Computación con número de registro B061544, adscrito a Centro de Investigación en Computación, manifiesta que es autora intelectual del presente trabajo de Tesis bajo la dirección de Dr. Alexander Gelbukh y cede los derechos del trabajo intitulado "Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección yledeneva@yahoo.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Yulia Nikolaevna Ledeneva

Nombre y firma

# *Abstract*

In the last two decades, an exponential increase in the available electronic information causes a big necessity to quickly understand large volumes of information. It raises the importance of the development of automatic methods for detecting the most relevant content of a document in order to produce a shorter text. Automatic text summarization (ATS) is an active research area dedicated to generate abstractive and extractive summaries not only for a single document, but also for a collection of documents. Other problems consist in resolve ATS in a language and dominion independent way.

In this thesis, we consider extractive text summarization for single and multi document tasks. As the first contribution of this thesis, we have identified that a typical extractive summarization method consists in four steps. First step is a term selection where one should decide what units will count as individual terms. The process of estimating the usefulness of the individual terms is called term weighting step. The next step denotes as sentence weighting where all the sentences receive some numerical measure according to the usefulness of its terms. Finally, the process of selecting the most relevant sentences calls sentence selection. Different extractive summarization methods can be characterized how they perform these steps.

As the main contribution, in the term selection step, we propose to detect multiword descriptions considering Maximal Frequent Sequences (MFSs), which bearing important meaning, while non-maximal frequent sequences (FSs), those that are parts of another FS, should not be considered. Our additional motivation was cost vs. benefit considerations: there are too many non-maximal FSs while their probability to bear important meaning is lower. In any case, MFSs represent all FSs in a compact way: all FSs can be obtained from all MFSs by bursting each MFS into a set of all its subsequences.

Other contributions are new methods based on graph algorithms, clustering algorithms, and genetic algorithms which facilitate the text summarization task. We have tested different combinations of term selection, term weighting, sentence weighting and sentence selection options for language and domain independent extractive single document text summarization on a news report collection. We analyzed several options based on multiword descriptions, considering them with graph, clustering, and genetic algorithms. We obtained results superior to the existing state-of-the-art methods. Also we tested the effect of pre-processing stage for obtaining multiword descriptions.

# Resumen

Un resumen del texto es un texto corto que transmite la información más importante del documento. El tipo de resúmenes con el cual trabajamos en esta tesis son resúmenes extractivos: una selección de las oraciones del texto más importantes. Se puede imaginar como la tarea de resaltar algunas oraciones del texto con un marcador amarillo; así que la tarea es decidir cuáles oraciones resaltar. Para eso, primero asignamos un valor numérico a las palabras o frases del documento; luego resaltamos las oraciones que contienen, en promedio, palabras o frases más importantes. Con eso, la tarea es determinar el peso de importancia de las palabras o frases. Más aún, la tarea es decidir cuales secuencias de palabras conviene considerar como frases. La tesis aborda una aproximación mayormente estadística, minimizando el conocimiento lingüístico manual requerido, entonces, por razones técnicas, nuestra noción de la frase significativa difiere de la aceptada en la lingüística tradicional. Se ha desarrollado una nueva noción del término multipalabra adecuada para nuestra tarea y ajustado para su tratamiento automático, y se ha comprobado su utilidad, alcanzando niveles mejorados de precisión en los resúmenes obtenidos.

Las contribuciones principales en la etapa de selección de términos que hemos propuesto es la detección de descripciones multipalabra considerando Secuencias Frecuentes Maximales (SFMs), cuales adquieren un significado importante mientras secuencias frecuentes (FS) no maximales los cuales son partes de otros FS, no deben de ser consideradas. En la motivación se consideró costo vs. beneficio: existe muchas FS no maximales mientras la probabilidad de adquirir un significado importante es baja. De todos modos, MFSs representan todas FSs en el modo compacto: todas FSs podrían ser obtenidas a partir de todas MFSs explotando cada MFS al conjunto de todas sus subsecuencias.

Otras contribuciones de este trabajo son nuevos métodos basados en grafos, algoritmos de agrupamiento, y algoritmo genético cuales facilitan la tarea de generación de resúmenes de textos. Se ha experimentado diferentes combinaciones de las opciones de selección de términos, pesado de términos, pesado de oraciones y selección de oraciones para generar los resúmenes extractivos de textos independiente de lenguaje y dominio para una colección de noticias. Se ha analizado algunas opciones basadas en descripciones multipalabra considerándolas en los métodos de grafos, agrupamiento y algoritmos genéticos. Se ha obtenido los resultados superiores al estado de arte. También se ha probado el efecto de pre-procesamiento para la obtención de resúmenes con los métodos propuestos.

# Contents

# List of Figures

# List of Tables

# *Chapter 1. Introduction*

## *1.1. Introduction*

A summary of a document is a (much) shorter text that conveys the most important information from the source document. There are a number of scenarios where automatic construction of such summaries is useful. For example, an information retrieval system could present an automatically built summary in its list of retrieval results, for the user to quickly decide which documents are interesting and worth opening for a closer look—this is what Google models to some degree with the snippets shown in its search results. Other examples include automatic construction of summaries of news articles or email messages to be sent to mobile devices as SMS; summarization of information for government officials, businessmen, researches, etc., and summarization of web pages to be shown on the screen of a mobile device, among many others.

Text summarization tasks can be classified into single document and multi document summarization. In single document summarization, the summary of only one document is to be built, while in multi document summarization the summary of a whole collection of documents (such as all today's news or all search results for a

query) is built. In this thesis we have experimented with single document and multi document summaries.

The summarization methods can be classified into abstractive and extractive summarization [Lin97]. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of "understanding" the original text and "re-telling" it in fewer words. Namely, an abstractive summarization method uses linguistic methods to examine and interpret the text and then to find new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original document. While this may seem the best way to construct a summary (and this is how human beings do it), in real-life setting immaturity of the corresponding linguistic technology for text analysis and generation currently renders such methods practically infeasible.

An extractive summary, in contrast, is a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order—i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. The resulting summary reads rather awkward; however, simplicity of the underlying statistical techniques makes extractive summarization an attractive, robust, language-independent alternative to more "intelligent" abstractive methods. In this paper, we consider extractive summarization.

A typical extractive summarization method consists in several steps, at each of them different options can be chosen. We will assume that the units of selection are sentences (these could be, say, phrases or paragraphs). Thus final goal of the extractive summarization process is *sentence selection*. One of the ways to select the appropriate sentences is to assign some numerical measure of usefulness of a sentence for the summary and then select the best ones; the process of assigning these usefulness weights is called *sentence weighting*. One of the ways to estimate the usefulness of a sentence is to sum up usefulness weights of individual terms of which the sentence consists; the process of estimating the individual terms is called

*term weighting*. For this, one should decide what the terms are: for example, they can be words; deciding what objects will count as terms is the task of *term selection*. Different extractive summarization methods can be characterized by how they perform these tasks.

In this thesis, we propose new term selection, term weighting, sentence weighting, and sentence selection steps. We analyze several options for simple language-independent statistical term selection and corresponding term weighting, based on units larger than one word. In particular, we are looking for new terms denoted as multiword descriptions which could be a good terms for the task of text summarization.

The thesis introduces basic concepts and definitions, summaries state-of-the-art of text summarization methods, describes proposed methods, and presents experimental settings and obtained results for different term selection, term weighting, sentence selection and sentence selection schemes. Preliminary conclusions are given.

## 1.2. Main Contributions

We mention some of the contributions that we obtained for the moment:

– Identification of general steps for an automatic extractive text summarization method.

– Proposal of new methods for generating text summarizes based on the discovery of multiword descriptions.

– Development of new methods for single document summarization.

– Development of new methods for multi document summarization

– New approaches to deal with the task of generation of summaries in a language independent way.

– New approaches to deal with the task of generation of summaries in a dominion independent way.

– Methods for automatic generation of text summaries which are superior to the state-of-the-art methods.

– One of the possible contributions is the application of multiword descriptions for other task of NLP.

## 1.3. Organization of the Document

This thesis is organized in seven chapters. This chapter introduces the work done in this dissertation. The next chapter summarizes state-of-the-art methods. In the third chapter preliminary concepts are given. In chapter 4 the proposed methods for single text summarization are presented. Chapter 5 describes the proposed methods for multi text summarization. In Chapter 6, the experimental results are reported and the final chapter presents the conclusions and a future work.

# CHAPTER 2. State-of-the-art

Mainly, this chapter is dedicated of the detailed presentation of the-state-of-the-art. We begin the first section with a detailed description of the state-of-the-art of extractive summarization. This section is sorted taking into account four steps of extractive summarization. Then, a brief state-of-the-art for abstractive summarization and applications of text summarization is given. Finally, we conclude the chapter exposing the description of the research problem, objectives and methodology of this dissertation.

## *2.1. State-of-the-art*

### *2.1.1. Extractive Text Summarization*

**Term Selection**

Most works discussed below are based on *words* as terms, however is not the only possible option. Liu et al. [Liu06a] uses pairs of syntactically connected words (*basic elements*) as atomic features (terms). Such pairs (which can be thought of as arcs in the syntactic dependency tree of the sentence) have been shown to be more precise semantic units than words [Kos04]. However, while we believe that trying text units larger than a word is a good idea, extracting the basic elements from the text requires dependency syntactic parsing, which is language-dependent. Simpler statistical methods (cf. the use of *ngrams* as terms in [Vil06]) may prove to be more robust and language-independent.

Some approaches of text summaries match semantic units such as *elementary discourse units* [Mar01, Sor03], *factoids* [Teu04a, Teu04b], *information nuggets* [Voo04], *basic elements* [Liu06a], etc. A big disadvantage of these semantic units is that the detection of these units is realized manually. For example, *information nuggets* are atomic pieces of interesting information about the target identified by human annotators as vital (required) or non-vital (acceptable but not required) for the understanding of the content of a summary.

*Factoids* are semantic units which represent the meaning of a sentence. For instance, the sentence "The police have arrested a white Dutch man" by the union of the following factoids: "A suspect was arrested", "The police did the arresting", "The suspect is white", "The suspect is Dutch", "The suspect is male". Factoids are defined empirically based on the data in the set of summaries (usually are some manually made summaries taken from [Duc]). Factoid definition starts with the comparison of the information contained in two summaries, and factoids get added or split as incrementally other summaries are considered. If two pieces of information

occur together in all summaries and within the same sentence, they are treated as one factoid, because differentiation into more than one factoid would not help us in distinguishing the summaries. Factoids are labeled with descriptions in natural language; initially, these are close in wording to the factoid's occurrence in the first summaries, though the annotator tries to identify and treat equally paraphrases of the factoid information when they occur in other summaries. If (together with various statements in other summaries) one summary contains "was killed" and another "was shot dead", we identify the factoids: "There was an attack", "The victim died", "A gun was used". The first summary contains only the first two factoids, whereas the second contains all three. That way, the semantic similarity between related sentences can be expressed. When factoids are identified in the collection of summaries, most factoids turned out to be independent of each other. But when dealing with naturally occurring documents many difficult cases appear, e.g. ambiguous expressions, slight differences in numbers and meaning, and inference.

The text is segmented in *Elementary Discourse Units* (EDUs) or non-overlapping segments, generally taken as clauses or clauses like units of a rhetorical relation that holds between two adjacent spans of text [Mar01, Car03]. The boundaries of EDUs are determined using grammatical, lexical, syntactic information of the whole sentence.

Other possible option proposed by Nenkova in [Nen06] is *Semantic Content Units* (SCUs). The definition of the content unit is somewhat fluid, it can be a single word but it is never bigger than a sentence clause. The most important evidence of their presence in a text is the information expressed in two or more summaries, or in other words, is the frequency of the content unit in a text. Other evidence is that these frequent content units can have different wording (but the same semantic meaning) what brings difficulties for language-independent solution.

The concept of *lexical chains* was first introduced by Morris and Hirst. Basically, lexical chains exploit the cohesion among an arbitrary number of related words [Mor91]. Then, lexical chains were computed in a source document by grouping (chaining) sets of words that are semantically related (i.e. have a sense flow) [Bar99, Sil02]. Identities, synonyms, and hypernym/hyponyms are the relations among words that

might cause them to be grouped into the same lexical chain. Specifically, words may be grouped when:

- Two noun instances are identical, and are used in the same sense. (The <u>house</u> on the hill is large. The <u>house</u> is made of wood.)
- Two noun instances are used in the same sense (i.e., are synonyms). (The <u>car</u> is fast. My <u>automobile</u> is faster.)
- The senses of two noun instances have a hypernym/hyponym relation between them. (John owns a <u>car</u>. It is a <u>Toyota</u>.)
- The senses of two noun instances are siblings in the hypernym/hyponyn tree. (The <u>truck</u> is fast. The <u>car</u> is faster.)

In computing lexical chains, the noun instances were grouped according to the above relations, but each noun instance must belong to exactly one lexical chain. There are several difficulties in determining which lexical chain a particular word instance should join. For instance, a particular noun instance may correspond to several different word senses and thus the system must determine which sense to use (e.g. should a particular instance of "house" be interpreted as sense 1: dwelling or sense 2: legislature). In addition, even if the word sense of an instance can be determined, it may be possible to group that instance into several different lexical chains because it may be related to words in different chains. For example, the word's sense may be identical to that of a word instance in one grouping while having a hypernym/hyponym relationship with that of a word instance in another. What must happen is that the words must be grouped in such a way that the overall grouping is optimal in that it creates the longest/strongest lexical chains. It was observed that contention that words are grouped into a single chain when they are "about" the same underlying concept. That fact confirms the usage of lexical chains in text summarization [Bru01, Zho05, Li07].

*Keyphrases*, also known as *keywords*, are linguistic units, usually, longer than a words but shorter than a full sentence. There are several kinds of keyphrases ranging from statistical motivated keyphrases (sequences of words) to more linguistically motivated ones (that are defined in according to a grammar). In keyphrases extraction task, keyphrases are selected from the body of the input document, without a predefined list. Following this approach, a document is treated as a set of

candidate phrases and the task is to classify each candidate phrase as either a keyphrase or nonkeyphrase [Dav07]. When authors assign keyphrases without a controlled vocabulary (free text keywords or free index terms), about 70% to 80% of their keyphrases typically appear somewhere in the body of their documents [Dav07]. This suggests the possibility of using author-assigned free-text keyphrases to train a keyphrases extraction system.

D'Avanzo [Dav07] extracts *syntactic patterns* using two ways. The first way focuses on extracting uni-grams and bi-grams (for instance, noun, and sequences of adjective and noun, etc.) to describe a precise and well defined entity. The second way considers longer sequences of part of speech, often containing verbal forms (for instance, noun plus verb plus adjective plus noun) to describe concise events/situations. Once all the uni-grams, bi-grams, tri-grams, and four-grams are extracted from the linguistic pre-processor, they are filtered with the patterns defined above. The result of this process is a set of patterns that may represent the current document.

For multi document summarization, *passages* are retrieved using a language model [Yin07]. The goal of language modeling is to predict the probability of natural word sequences; or in other words, to put high probability on word sequences those actually occur and low probability on word sequences that never occur. The simplest and most successful basis for language modeling is the n-gram model.

**Term Weighting**

Of the works devoted to term-based methods, most concentrate on term weighting. Terms identified in the previous step are scored in order to select the most appropriate terms as representative of the original text.

The use of frequency as a feature in text summarization has been proven useful. Term frequency was first used in extractive text summarization in the late 1950's [Luh58]. Subsequent research using frequency methods focused on the use of frequency as one feature among many for identifying important terms [Edm99]. Most recently, the SumBasic algorithm uses term frequency as part of a context-sensitive approach to

identifying important sentences while reducing information redundancy [Nen05b]. See more about frequency weighting in the section of *Sentence Selection*.

The proposed scoring by D'Avanzo [Dav07] is based on a combination of *tf* x *idf* and first occurrence, i.e. the distance of the candidate term (or specifically keyphrases are used as terms) from the beginning of the document in which it appears. However, since candidate phrases do not appear frequently enough in the collection, it has been decided to estimate the values of the *tf* x *idf* using the head of the candidate phrase, instead of the whole term. Every phrase has a single word as head. The head is the main verb in the case of verb phrases, and a noun (last noun before any post-modifiers) in noun phrases. As learning algorithm, it has been used an SVM. The classifier was trained on a corpus with the available keyphrases. From the document collection we extracted all nouns and verbs. Each of them was marked as a positive example of a relevant keyphrase for a certain document if it was present in the assessor's judgment of that document; otherwise it was marked as a negative example. Then the two features (i.e. *tf* x *idf* and first occurrence) were calculated for each word. The classifier was trained using this material and a ranked word list was returned. The system automatically looks in the candidate phrases for those phrases containing these words. The top candidate phrases matching the word output of the classifier are kept. The model obtained is reused in the subsequent steps. When a new document or corpus is ready we use the pre-processor module to prepare the candidate phrases. The model we got in the training is then used to score the phrases obtained. In this case the pre-processing part is the same. So, using the model we got in the training, we extract nouns and verbs from documents, and then we keep the candidate phrases containing them. The system from [Dav07] uses two parameters for controlling its work: one is the maximum number of words allowed in a keyphrase and the second is the maximum number of keyphrases to be extracted from a document.

Nenkova et al. [Nen04, Pas05, Nen06] annotates special terms using the pyramid scheme—a procedure specifically designed for comparative analysis of the content of several texts. The idea of this scheme is to calculate presence of each term in all documents of the collection. The more documents have the term, the more important is this term, and consequently will be included in the summary.

Wei et al. [Wei06] derives relevance of a term from an ontology constructed with formal concept analysis. Song et al. [Son04] basically weight a word basing on the number of lexical connections, such as semantic associations expressed in a thesaurus, that the word has with its neighboring words; along with this, more frequent words are weighted higher. Mihalcea [Mih06] presents a similar idea in the form of a neat, clear graph-based formalism: the words that have closer relationships with a greater number of "important" words become more important themselves, the importance being determined in a recursive way similar to the PageRank algorithm used by Google to weight webpages.

The latter idea can be applied directly to sentence weighting without term weighting: a sentence is important if it is related to many important sentences, where relatedness can be understood as, say, overlap of the lexical contents of the sentences [Mih06]. The two methods presented in [Mih06] are those that currently give the best results and with which we compare our suggested method.

**Sentence Weighting**

Ideally, a text summarization system should "understand" (analyze) the text and express its main contents by generating the text of the summary. For example, Cristea et al. [Cri05] perform sentence weighting according to their proximity to the central idea of the text, which is determined by analysis of the discourse structure.

However, the techniques that try to analyze the structure of the text involve too sophisticated and expensive linguistic processing. In contrast, most of the methods discussed in the literature nowadays represent the text and its sentences as a bag of simple features, using statistical processing without any attempts to "understand" the text.

A very old and very simple sentence weighting heuristic does not involve any terms at all: it assigns highest weight to the first sentences of the text. Texts of some genres—such as news reports or scientific papers—are specifically designed for this heuristic: e.g., any scientific paper contains a ready summary at the beginning. This gives a

baseline [DUC] that proves to be very hard to beat on such texts. However, comparing term-based methods with such position-based baseline is not fair in the sense that this baseline only works on text of specific genres (say, it will not work on official documents, email messages, webpages, or literary novels) and uses information (the position of the sentence) not available to term-based methods. It is worth noting that in Document Understanding Conference (DUC) competitions [DUC] only five systems performed above this baseline, which does not demerit the other systems because this baseline is genre-specific. Though the method proposed in this paper very slightly outperforms this baseline, such a comparison is unfair.

Another of the possible approaches is *relative utility* [Rad03]. In this approach, all sentences in the input are ranked on a scale from 0 to 10 as to their suitability for inclusion in a summary. In addition, sentences that contain similar information are explicitly marked, so that in the evaluation metric one could penalize for redundancy and reward equally informationally equivalent sentences. The ranking of sentences from the entire input allows for a lot of flexibility, because summaries of any size or compression rate can be evaluated. At the same time, the method is applicable only to extractive systems that select sentences directly from the input and do not attempt any reformulation or regeneration of the original journalist-written sentence. The relative utility approach is very similar in spirit to the evaluation used by [Mar00], who asked multiple independent subjects to rank the importance of information units. The main difference is that earlier research directly concentrated on subsentential units rather than sentences.

Verma [Ver07] utilizes ontology knowledge for weighting sentences using statistical data of sentences and also parsing and syntax analysis. The disadvantage of this proposal is that was maid for only one particular domain.

**Sentence Selection**

Supervised learning methods consider sentence selection as a classification task: they train a classifier using a collection of documents supplied with existing summaries. As features of a sentence such methods can consider text units (in which case we can speak of term selection) or other, non-lexical characteristics.

Different lexical and non-lexical features have been used in [Kup95, Chu04, Net04]. Most of these features are "heuristically motivated", since they tend to emulate the manual creation of extracts. In a work of Kupiec [Kup95], the following features were proposed: *sentence position*, *sentence length*, *the presence of key phrases* and *overlap with the title of the document*. More recent works [Chu04, Net04] extend these features incorporating information about the occurrence of proper names and the presence of anaphors. The "heuristically motivated" features allow extract very precise summaries. However, they have a very big disadvantage of being highly linked to a specific domain. This condition implies that the change for one domain to another, it may be necessary to redefine or even eliminate some features. For instance, key phrases, which are particular for each domain, require being modified, while the overlap with the title, which has no sense in all topics, may be eliminated.

In order to increase the domain (and language) independence of machine learning summaries, Villatoro [Vil06] eliminates all kind of "heuristically motivated" attributes and substitute them by word-based features. In particular, he uses word sequences (*ngrams*) as terms. Although the first attempt to use ngrams is exceeded the results of other methods, it has some disadvantages. One is that they are always sequences of a fixed size, which was previously defined by the user. The big part of the problem in such techniques lies in defining the size of the sequence to be extracted, which usually depends on the analysis of the text.

Chali and Kolla [Cha04, Kol05] presented work on multi document summarization using lexical chains for sentence selection. In [Cha04], the scoring mechanism only considers the number of occurrences of words within a sentence and within a segment (segment in their work is comparable to single documents within a collection for one topic). No additional information like n-grams is used. The obtained results report that sentence scoring based on simple lexical chain counts is not performing enough. As a continuation, the scoring was changed in [Fil07] adding different scoring. One added score was the number of chains passing through a sentence (score-chain) and the other score was based on n-grams (scorebigram, scoretrigram). Each occurrence of a chain and bigram increased the score by 1; of a trigram increased the score by 2. The overall score was calculated for each

sentence and then used to rank sentences for the final extraction. Using the number of chains passing through a sentence gives higher scores to longer sentences as they are likely to have several chains passing through. But longer sentences also have a higher likelihood to contain more information—especially, if several chains pass through them. The score based on n-grams finds sentences that also have a high score based on identity. It also emphasizes sentences where words from the topic not only occur somewhere in the sentences, but also where words occur in the same order as in the topic sentences.

The approach from [Sek02] is based on weighting approach calculated in the following way. First, the *tf x idf* values of all nouns in the document except some stop words are computed. According to each document, the sum of all the *tf x idf* values of nouns in the document is computed. The importance value of a sentence is computed by the sum of *tf x idf* values of sentences containing nouns divided by the sum of all *tf x idf* values in the document. Second, if a sentence contains phrases in the heading, the number of phrases is divided by the total phrases in the heading. That value is then multiplied by the constant 0.1, and adds to the sentence weights. Third, the line number of the sentence in the document divided by the number of all lines in the document corresponds to the position value in the document from 0 to 1. And thus the resultant importance value of each sentence is obtained.

Extractive approaches to text summarization usually follow a model of scoring sentences based on a set of features. The highest scoring sentences are then extracted to form a summary. When using frequency as the only feature, unit items are counted and then each sentence is given a score based on the frequency count of each unit item in the sentence. A key problem in generating summaries is reducing redundancy. Each new sentence in the summary should add new information rather than repeating already included information. Using the highest frequency terms will likely result in the same information repeatedly being selected, with the chance that some additional information is included. In the SumBasic [Nen05b] frequency approach, a probability distribution model is first generated, and as each term is used to select sentences, the term probabilities are reduced so that lower probability terms have a better chance of selecting sentences with new information content. This approach is called context sensitive since the summarizer

considers sentences already in the summary before selecting a new sentence to add to the summary. This is also related to the idea of finding Maximal Marginal Relevance (MMR), where marginal relevance is defined as finding relevant sentences which contain minimal similarity to previously selected sentences [Car98].

The frequency distribution algorithm—FreqDist—uses a context sensitive approach to scoring sentences based on a frequency distribution model rather than a probability distribution model [Ree07]. The rationale of the frequency distribution approach is that the frequency distribution of terms or concepts in the source text ought to appear in the generated summary as closely as possible to the source text. That is, the frequency distribution models of the source text and its corresponding summary should be as similar as possible. There are two stages in the algorithm: Initialization and Summary Generation. In the initialization stage, the unit items (terms or concepts) of the source text are counted to form a frequency distribution model of the text, and a pool of sentences from the source text is created, called the sentence pool. A summary frequency distribution model is created from the unit items found in the source text. The summary frequency distribution model frequency counts are initially set to zero to indicate an empty summary. In the Summary Generation stage, new sentences are evaluated and then selected for inclusion in the summary. Identifying the next sentence to be added to the summary is accomplished by finding the sentence which most closely aligns the frequency distribution of the summary generated so far to the frequency distribution of the original source text. A candidate summary is first initialized to the summary generated so far. For each sentence in the sentence pool, the sentence is added to the candidate summary to see how much it contributes to the candidate summary. To determine the sentence's contribution, the candidate summary frequency distribution is compared for similarity to the source text's frequency distribution. The comparison generates a similarity score. This similarity score is assigned to the sentence as the sentence's score. After all sentences from the sentence pool have been evaluated for their contribution to the candidate summary, the highest scoring sentence is added to the summary. The sentence added to the summary is then removed from the sentence pool. The sentence selection process is iterative, and repeats until the desired length of the summary is reached.

However, the majority of current methods are purely heuristic: they do not use any learning but directly state the procedure used for term selection, term weighting, and/or sentence weighting (given that sentence selection in most cases consists in selecting the best-weighted sentences).

## 2.1.2. Abstractive Text Summarization

Abstractive summarization approaches use information extraction, ontological information, information fusion, and compression. Automatically generated abstracts (abstractive summaries) moves the summarization field from the use of purely extractive methods to the generation of abstracts that contain sentences not found in any of the input documents and can synthesize information across sources. An abstract contains at least some sentences (or phrases) that do not exist in the original document. Of course, true abstraction involves taking the process one step further. Abstraction involves recognizing that a set of extracted passages together constitute something new, something that is not explicitly mentioned in the source, and then replacing them in the summary with the (ideally more concise) new concept(s). The requirement that the new material not be in the text explicitly means that the system must have access to external information of some kind, such as an ontology or a knowledge base, and be able to perform combinatory inference.

Different methods are developed in abstractive summarization. For example, techniques of sentence fusion [Dau04, Bar03, Bar05], information fusion [Bar99], sentence compression [Van04, Mad07], etc.

## 2.1.3. Applications of Text Summarization

We can find different systems made for the summarization of the following applications:
- Legal texts [Far04, Har04].
- Emails [Cor04, Shr04, Wan04].
- Web pages [Dia06].
- Web documents using mobile devices [Ott06].

- Figures and graphics [Fut04, Car04, Car06].
- News [Eva05, Mck03, Nen05a].

## *2.2. Research Problem*

The big number of researches in CL caused the development of more efficient algorithms. This influenced that the usage of text summarization has grown considerably in many of the scientific and business applications in this days. In the great majority of the cases, text summarization methods generate good summaries based on the language- and domain- dependent techniques, nevertheless these summaries are made generally with a low quality in the content compared to the ability shown by a not trained person. In order to solve this problem, text summarization methods made possible that the user had tools to summarize texts, however a data interchange in many languages and domains increases the necessity in complex design of new methods. The implementation of such types of automatic text summarization methods has become more difficult and eventually impossible due to the natural language complexity.

The limitations caused by the complexity of natural language, suggested new methodologies for the design of text summarization methods through the use of language and domain- dependent methods. Although such methods denote to be the best option to reduce text dimensions, some of these methods include some specific problems such that much content is not relevant for the general understanding of complete texts. The design of these methods includes the composition of summaries which in the state-of-the-art methods depends on the extraction of the units without considering their importance as descriptors of a text.

Here arises the following research question: How to automatically detect the most important parts of a text for composing a summary in a language and dominion independent way?

The purpose of the present work is to present new methods for producing a text summary extracting the terms which describe the most important information of the

text. The proposed methods will generate summaries in a language and dominion independent way. These methods will include different steps, adjusting each of them for the improvement of the total results. The corpus with manually elaborated summaries is used to exemplify and to verify the effectiveness of the methods.

Finally, we will mention the cases where we can use summarization:

1. Selecting scientific papers and thesis if it talks about the topic we are working on. Including multi document summarization for composing the state-of-the-art of a topic.
2. Before we buy a book, generally we read a brief description of it.
3. As students, we prefer to ask the professor for a class's content until we sign in his class.
4. Short description of TV, radio, entertainment programs.
5. Mainly, navigating and looking for the information in www.

## 2.3. Objectives

1. New state-of-the art methods in various tasks of automatic generation of summaries of a single document and a collection of documents, with application to a different languages.
2. Development of the system for automatic generating of summaries with the superior quality to the state-of-the art methods. The system will be useful for the users and also will serve as a framework and evaluation for the developed methods.
3. New approaches to deal with the task of generation of summaries in a language and dominion independent way.
4. New approaches for generating text summarizes based on the discovery of multiword descriptions.

## *2.4. Methodology*

For the development of new methods for automatic generation of text summaries, it is necessary to realize the following methodology:

1.  Pre-processing stage:
    1.1.    Preparation of corpus for the stage of experiments.
    1.2.    Usage of lexical resources of the-state-of-the-art (WordNet, EuroWordNet, WordNet Similarity, GATE (General Architecture for Text Engineering), Natural Language Toolkit).
    1.3.    Usage of measures for the evaluation of proposed methods: precision, recall, f-measure).
    1.4.    Implementation of the methods and tools for the evaluation of proposed methods (ROUGE, SEE).
    1.5.    Development in programming languages (Java, Perl, Builder C++).

2.  Automatic generation of summaries of the-state-of the-art methods:
    2.1.    Extraction of MFSs for single document.
    2.2.    Extraction of MFSs for collection of documents.
    2.3.    Generation of summaries for different languages.
    2.4.    Proves, adjustments, documentation.

3.  Automatic generation of summaries:
    3.1.    Development of methods for automatic generation of text summaries which includes:
        o   Extraction of multiword descriptions of single document.
        o   Extraction of multiword descriptions of collection of documents.
    3.2.    Implementation of the methods for automatic generation of summaries for English language.
    3.3.    Implementation of the methods for automatic generation of summaries for different languages.
    3.4.    Proves, adjustments, documentation.

4. Documentation and presentations.
    4.1.    Publishing of papers.
    4.2.    Presentation of papers in the conferences.

This chapter has an objective to provide the basic concepts necessary to the comprehension of this thesis.

We begin this chapter with a presentation of natural language Laboratory where this dissertation is written. Section 3.1.1 describes the area of Computational Linguistics and its applications. In section 3.1.2, we can find an introduction to Text Summarization and evaluation measures used for summarization. Then, text pre-processing and test representation models are explained.

## 3.1. Computational Linguistics

The area of Computational Linguistics (CL) describes the modern models of how natural language processing systems function, explains how to compile the data for the necessity for the Natural Language Processing (NLP) systems: develops software for grammas correction, resolves word sense disambiguation, constructs dictionaries and databases, retrieves information, translates automatically from one language to another, etc. [Bol04a]. As many other areas (for example, such as mechanical and engineering areas), CL has the necessity of intelligent language processing tools and automation of NLP tasks.

We mention some of the areas of NLP:

- Word Sense Disambiguation (WSD) [Gel03a, Man99]: solves what sense has a given word, generally based on its context. This task is very important because of its successful resolution, depends the correctness of other applications such as Machine Translation, Question Answering, etc.
- Information Retrieval (IR) [Man07, Bae99]: consists of finding documents of an unstructured nature that satisfies an information need from within large collections of documents usually on local computer or on the internet. This area overtakes traditional database searching, becoming the dominant form of information access. Now hundreds of millions of people use IR systems every day when they use a web search engine or search their emails.
- Machine Translation [Gel03b, Bol04a]: is a machine-assisted system responsible for translation from one language to another. This application is very useful for bossiness and scientific purposes by reason that the international collaboration grows exponentially.
- Question Answering (QA) [Ace07, Fer07]: is a complex task that combines techniques from NLP, IR and machine learning. The main aim of QA is to localize the correct answer to a question written in natural language in a non-structured collection of documents. Systems of QA look like a search engine

where the input to the system is a question in natural language and the output is the answer to the question (not a list of entire documents like in IR).

## 3.1.1. Computational Linguistics in Mexico

More than 50 years have passed since first advances were published in the area of CL. From this time a lot of work has been done all over the world including Mexico. Especially, I would like to mention a work of Natural Language Laboratory where more than 300 papers during last 10 years were created mainly by three researches: Ph.D. Alexander Gelbukh, Ph.D. Igor Bolshakov, and Ph.D. Grigori Sidorov [Gel08]. Their works establish basic definitions and new research discoveries in different tasks of CL:

- Lexical Resources [Gel06]
- Construction and Compilation of Dictionaries [Gel02; Gel03c; Gel04a]
- Database of Collocations (CrossLexica) [Bol01; Bol04b; Bol08]
- Syntactic Analysis of the Spanish Language [Gal07]
- Semantic Errors and Malapropism [Gel04b; Bol05]
- Word Sense Disambiguation [Gel03a; Ledo03]
- Automatic Translation [Gel03b]
- Text Mining [Mon01; Mon02]

It is honorable to point out that the one of the main conference of CL is organized in this Laboratory by Prof. Gelbukh [CICLing].

Since the foundation of this Laboratory, CL has grown to become a major scientific domain in Mexico. For the past decade, educational and commercial CL systems have been successfully developed [Sid05] (see [Gel08] for more details). Interest of scientists of all over the world has also been growing, as we can see in special organized conferences [CICLing], and publishing plenty of works [Gel08]. Moreover, several Ph.D. students gradated from the Laboratory establish new Natural Language Laboratories [Ina08, Una08]. This dissertation is one of the contributions in the unlimited space of CL.

## 3.2. Text Summarization

Early experimentation in the late 1950s and 1960s suggested that text summarization by computer was feasible though not straightforward. After a gap of some decades, progress in language processing, coupled with the growing presence of on-line text—in corpora and especially on the web—renewed interest in automated text summarization. So, the huge amount of available electronic documents in Internet has motivated the development of very good information retrieval systems. However, the information provided by such systems, like *google*, only show part of the text where the words of the request query appears. Therefore, the user has to decide if a document is interesting only with the extracted part of a text. Moreover, this part does not have any information if the retrieved document is interesting for the user, so it is necessary download and read each retrieved document until the user finds satisfactory information. It was unnecessary and time-consuming routine. A solution for such problems is to achieve an automatic text summarization of the document extracting the essential sentences of the document.

The demand of the automatic generation of text summaries has appeared in other areas, for example, summaries of news articles; summaries of electronic mails and news to send them as SMS; summaries of information (for government officials, businessmen, researches, etc.); summaries of web pages to transmit them through telephone; in searching systems to receive the summaries of found documents and pages.

From one side, there is a single document summarization which implies to communicate the principal information of one specific document, and from another side—a multi document summarization which transmits the main ideas of a collection of documents. There are two options to achieve a summarization by computer: text abstraction and text extraction [Lin97]. Text abstraction examines a given text using linguistic methods which interpret a text and find new concepts to describe it. And then new text is generated which will be shorter with the same content of information. Text extraction means extract parts (words, sequences, sentences, paragraphs, etc.) of a given text based on statistic, linguistic or heuristic methods,

and then join them to new text which will be shorter with the same content of information.

According to the classical point of view (see below how we introduce our point of view), there are three stages in automated text summarization [Hov03]. The first stage is performed by *topic identification* where almost all systems employ several independent modules. Each module assigns a score to each unit of input (word, sentence, or longer passage); then a combination module combines the scores for each unit to assign a single integrates score to it; finally, the system returns the *n* highest-scoring units, according to the summary length requested by the user. The performance of topic identification modules is usually measured using Recall and Precision scores (see Section below).

The second stage denotes as the stage of *interpretation*. This stage distinguishes extract-type summarization systems from abstract-type systems. During the interpretation the topics identified as important are fused, represented in new terms, and expressed using a new formulation, using concepts or words not found in the original text. No system can perform interpretation without prior knowledge about the domain; by definition, it must interpret the input in term of something extraneous to the text. But acquisition deep enough prior domain knowledge is so difficult that summarizers to date have only attempted it in a small way. So, the disadvantage of this stage remains blocked by the problem of domain knowledge acquisition.

Summary *generation* is the third stage of text summarization. When the summary content has been created in internal notation, and thus requires the techniques of natural language generation, namely text planning, sentence planning, and sentence realization.

We identified four steps for composing a text summary:

- *Term selection*: during this step one should decide what units will count as terms are, for example, they can be swords, ngrams or phrases.
- *Term weighting*: this is a process of weighting (or estimating) individual terms.

- *Sentence weighting*: the process of assigning numerical measure of usefulness to the sentence. For example, one of the ways to estimate the usefulness of a sentence is to sum up usefulness weights of individual terms of which the sentence consists.
- *Sentence selection*: selects sentences (or other units selected as final parts of a summary). For example, one of the ways to select the appropriate sentences is to assign some numerical measure of usefulness of a sentence for the summary and then select the best ones.

## *3.2.1. Evaluation of Summaries*

**Evaluation using ROUGE**:

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin03a] was proposed by Lin and Hovy [Lin04a, Lin04b, Lin04c]. This system calculates the quality of a summary generated automatically by comparing to the summary (or several summaries) created by humans.[1] Specifically, it counts the number of overlapping different units such as word sequences, word pairs and ngrams between the computer-generated summary to be evaluated and the ideal summaries created by humans. ROUGE includes several automatic evaluation measures:

- ROUGE-N (ngrams co-occurrence):
  Is an ngram recall between a candidate summary and a set of reference summaries, and calculates as follows:

$$ROUGE - N = \frac{\sum\limits_{S \in \{\mathrm{Re}\, ferenceSummaries\}} \sum\limits_{gram_n \in S} count_{match}(gram_n)}{\sum\limits_{S \in \{\mathrm{Re}\, ferenceSummaries\}} \sum\limits_{gram_n \in S} count(gram_n)},$$

  where $n$ is a length of the *ngram*, $gram_n$ and $count_{match}(gram_n)$ is the maximum number of ngrams co-occurring in a candidate summary and a set of reference summaries.

- ROUGE-L (longest subsequence):

---

[1] There is only some corpus which has summaries made by humans. It is very time-consuming and tedious work composing summaries manually. See Section 5.1 for more corpus details.

A sequence $S = (s_1, s_2, ..., s_n)$ is a subsequence of another sequence $X = (x_1, x_2, ..., x_m)$, if there exists a strict increasing sequence $(i_1, i_2, ..., i_k)$ of indices of $X$ such that all $j = 1, 2, ..., k$, than $x_{ij} = s_j$ [Cor89]. 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 in summarization evaluation, a summary sentence viewed as a sequence of words. Intuitively, the longer the LCS of two summary sentences is, the more similar the two summaries X of length m and Y of length n, assuming X is a reference summary sentence and Y is a candidate summary sentence.

– ROUGE-W (weighted longest subsequence):
Given two sequences $X$ and $Y$, LCS is called weighted if a length is calculated using a weighted function. For more details about weighted function see in [Lin03a].

– ROUGE-S (skip-bigram co-occurrence):
Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between a candidate summary and a set of reference summaries.

It is showed in [Lin03b] that these types of measures can be applied for evaluating the quality of summaries generated automatically achieving 95% of correlation of human judgments.

For each of the measures (ROUGE-N, ROUGE-L, ROUGE-W, etc.), ROUGE returns Recall, Precision and F-measure scores as follows:

Precision (*P*): reflects how many of the system's extracted sentences were good

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

Recall (*R*): reflects how many good sentences the system missed

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

F-measure (*F*):

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

where *correct* is the number of sentences extracted by the system and the human, *wrong* is the number of sentences extracted by the system but not by the human, and *missed* is the number of sentences extracted by the human but not by the system.

## 3.3. Text Pre-processing

The pre-processing step is perhaps the most important in the area of computational linguistics, since the quality of the obtained summary depends on how efficient is the representation of a text. In this thesis, some experiments will contain the pre-processing stage. Generally, this stage will include only two steps: eliminating stopwords and applying stemming.

### 3.3.1. Stopwords

When a pre-processing of a text is realized, an intermediate representation of it is obtained. One of the pre-processing stages consists in eliminating stopwords or empty words from the text. There is a set of empty words in every language, common to all domains which are easily identified, for example, articles, prepositions, conjunctions, etc. Although they can be verbs, adjectives and adverbs.

The words that are too frequent in the documents in a particular collection are not good discriminators. In fact, it is considered that a word which appears in at least 80% of the documents of a particular collection is useless for purpose of retrieval.

These words are considered empty and normally are removed to avoid being considered as potential.

Later, we will realize a process of extraction of stop-words in the documents with the aim of reducing the content of the text to more specific expressions (we call them multiword descriptions), containing only the words that are useful and meaningful for the generation of automatic summaries.

### 3.3.2. Stemming

Stemming technique consists in obtaining the root of words, so that the text processing is conducted on the roots and not on the original words. This technique allows to relate more terms in the document. It is supposed that two words that have the same root represent the same concept. Basically, the process of stemming of the words is realized for reducing to a minimum common portion of a word called stem. The stem is the portion of the word which is left when after the removal of its affixes, prefixes and suffixes. Once implemented stemming, the document will contain only the roots of the words. This will simplify the representations of the documents using the models mentioned above and clustering methods.

The first stemming algorithm was developed for the English language, and then was adapted for the Spanish language. The algorithm Porter [Por80] is the most commonly used for the English language. Also there are algorithms for other languages such as French, Dutch, Greek and Latin. In general, these algorithms are based on a simple set of rules that cut off words to obtain a common root [Bae99].

## 3.4. Text Representation Models

Text representation models are techniques based on the extraction of terms of a text or document which consist in choosing terms that will be extracted and than turned into terms. The difference between models are the type of terms that are extracted

from a document. In this thesis, three models are considered: bag of words (proposed by Salton in 1975), ngrams and MFSs.

## *Bag of Words*

The representation with bag of words consists in obtaining all different words which appear in a text. Subsequently, the document is shown as a vector, where each position will contain a term, and each term will correspond to a word in the document.

## *Ngrams*

The ngram model follows the same principle that the model based on bag of words, likewise the text is represented as a vector of terms. The difference is that the size of ngram is previosly defined, i.e. *n* is the number of consecutive elements that contain the term. These elements can be words or characters. For example, if *n* equals 2, the defined term will contain 2 words or characters, namely bigrams.

Observe that the n-gram model, extracted elements not completely preserve the order they appear in the text. In addition, we find another disadvantage common to both models: high dimensionality. Clearly, even with a small document, you have a considerable amount of different characteristics to evaluate, which means an enormous expense of resources to handle such amount of information. Trying to solve the problems of sequential order and dimensionality of models, MFSs has been proposed to use as a model representation of the text.

## *MFSs*

Frequent Sequence (FS) is a sequence of words or characters that appear in a text for the repeated manner. A sequence is called a maximal frequent sequence if it is not contained in another FS. MFSs model determines the number of times the FS will be repeated in the text to be considered frequent. This number is called the threshold. For example, for the document presented in Figure 3.1, Boolean model

based on MFSs is shown, taking as a threshold equal to 2. This means that each sequence must be appeared at least 2 times in the document to be frequent. Table 3.1 shows the representation of boolean MFSs model.

*…El gobierno de Egipto protege las pirámides…*
*…Las pirámides de Egipto son un patrimonio cultural…*
*…Las pirámides fueron construidas por los faraones…*
*…Las pirámides de Egipto fueron tumbas para los faraones de Egipto...*
*…Un buen gobierno protege su patrimonio cultural...*

Figure 3.1 – Example of 5 sentences from an arbitrary text.

Table 3.1 – Boolean representation of MFSs model for the example from Figure 3.1.

| MFSs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| gobierno | 1 | 0 | 0 | 1 | 0 |
| las pirámides de Egipto | 0 | 1 | 0 | 0 | 1 |
| patrimonio cultural | 0 | 1 | 0 | 1 | 0 |
| los faraones | 0 | 0 | 1 | 0 | 1 |

As shown in Table 3.1, the number of terms for MFSs model is reduced considerably comparing to bag of words and ngrams models. Tables 3.2, 3.3, 3.4, and 3.5 show models based on MFSs using 4 different weightings.

Table 3.2 – Representation of MFSs model with Boolean weighting.

| MFSs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| gobierno | 1 | 0 | 0 | 1 | 0 |
| las pirámides de Egipto | 0 | 1 | 0 | 0 | 1 |
| patrimonio cultural | 0 | 1 | 0 | 1 | 0 |
| los faraones | 0 | 0 | 1 | 0 | 1 |

Table 3.3 – Representation of MFSs model with *tf* weighting.

| MFSs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| gobierno | 1 | 0 | 0 | 1 | 0 |
| las pirámides de Egipto | 0 | 1 | 0 | 0 | 1 |
| patrimonio cultural | 0 | 1 | 0 | 1 | 0 |
| los faraones | 0 | 0 | 1 | 0 | 1 |

Table 3.4 – Representation of MFSs model with *idf* weighting.

| MFSs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| gobierno | 0.397 | 0 | 0 | 0.397 | 0 |
| las pirámides de Egipto | 0 | 0.397 | 0 | 0 | 0.397 |
| patrimonio cultural | 0 | 0.397 | 0 | 0.397 | 0 |
| los faraones | 0 | 0 | 0.397 | 0 | 0.397 |

Table 3.5 – Representation of MFSs model with *tf-idf* weighting.

| MFSs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| gobierno | 0.397 | 0 | 0 | 0.397 | 0 |
| las pirámides de Egipto | 0 | 0.397 | 0 | 0 | 0.397 |
| patrimonio cultural | 0 | 0.397 | 0 | 0.397 | 0 |
| los faraones | 0 | 0 | 0.397 | 0 | 0.397 |

As can be seen, in some cases the tables are identical, it can be explained by simplicity of the example, the values do not always change with respect to each sentence from one type weighting to another.

## *Term Weighting*

Within vector space models there is another way of representing a document which is denoted as term weighting. Term weighting consists in assigning a weight for each term which reflects the importance of the term in the document. Below 4 different term weighting options are described.

**Boolean Weighting**. It's the easiest way to weigh a term. It has 1, if it appears in the document and 0 in another case.

$$p_i(t_j) = \begin{cases} 0, & if \quad appeared \\ 1, & other \quad case \end{cases} \tag{3.1}$$

where $p_i(_j)$ is term frequency *j* in the document *i*.

**Term Frequency (t*f*)** was proposed in [Luh57]. This weighting takes into account that a term that frequently occurs in a document can better reflect the contents of the document than a term that occurs less frequent. Therefore, the weighting *tf* assigns a greater weight to terms with greater frequency and consists in evaluating the number of times when the word occurs in the document.

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

where $f_{ij}$ is term frequency *j* in the document *i*.

**Inverse Document Frequency (id*f*)** was proposed in [Sal88]. Taking into account the observation that a very frequent term appeared in several documents is less useful than a term that is appeared less frequent, because evaluates the distribution of terms in the document. The inverse frequency of the document is defined as:

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

where $f_{ij}$ is term frequency *j* in the document *i*; *N* is the number of the documents in the collection; $n_j$ is the number of documents where the term *j* appears.

**t*f*-id*f* Weigthing**. It is common that term frequency (t*f*) and inverse term frequency (id*f*) of the document are used together in order to determine the weight of each term in the vector space model [Sal88]. This combination is known as weighting t*f*-id*f* and consists of multiplying the frequency of the term by the inverse frequency of the documents where appears this term.

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

Note that in this thesis, if we are working with a single document, we will take *N* as the number of sentences y $n_j$ as the number of sentences where the term appears.

## 3.5. Graph Algorithms

Many language processing applications can be modeled by means of a graph. These data structures have the ability to encode in a natural way the meaning and

structure of a cohesive text, and follow closely the associative or semantic memory representations.

Particularly relevant in this thesis is the application of random-walks to text processing, as done in the TextRank system [Mih06]. TextRank has been successfully applied to three natural language processing tasks: document summarization, word sense disambiguation, and keyword extraction, with results competitive with those of state-of-the-art systems. The strength of the model lies in the global representation of the context and its ability to model how the co-occurrence between features might propagate across the context and affect other distant features. The description of TextRank is given below.

## 3.5.1. Graph Representation of Text

To enable the application of graph-based ranking algorithms to natural language texts, a graph that represents the text is built, and interconnects words or other text entities with meaningful relations. The graphs constructed in this way are centred around the target text, but can be extended with external graphs, such as off-the-shelf semantic or associative networks, or other similar structures automatically derived from large corpora.

**Graph Nodes:** Depending on the application at hand, text units of various sizes and characteristics can be added as vertices in the graph, e.g. words, collocations, wordsenses, entire sentences, entire documents, or others. Note that the graph-nodes do not have to belong to the same category.

**Graph Edges:** Similarly, it is the application that dictates the type of relations that are used to draw connections between any two such vertices, e.g. lexical or semantic relations, measures of text cohesiveness, contextual overlap, membership of a word in a sentence, and others.

**Algorithm:** Regardless of the type and characteristics of the elements added to the graph, the application of the ranking algorithms to natural language texts consists of the following main steps:

1. Identify text units that best define the task at hand, and add them as vertices in the graph.
2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.
3. Apply a graph-based ranking algorithm to find a ranking over the nodes in the graph. Iterate the graph-based ranking algorithm until convergence. Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

## 3.5.2. Graph Ranking Algorithms

The basic idea implemented by a random-walk algorithm is that of "voting" or "recommendation." When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

Moreover, the importance of the vertex casting a vote determines how important the vote itself is, and this information is also taken into account by the ranking algorithm. While there are several random-walk algorithms that have been proposed in the past, we focus on only one such algorithm, namely PageRank [Bri98], as it was previously found successful in a number of applications, including Web link analysis, social networks, citation analysis, and more recently in several text processing applications.

Given a graph $G = (V, E)$, let $In(V_i)$ be the set of vertices that point to vertex $V_i$ (predecessors), and $Out(V_i)$ be the set of vertices that vertex $V_i$ points to (successors). The PageRank score associated with the vertex $V_i$ is defined using a recursive function that integrates the scores of its predecessors:

$$S(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{S(V_j)}{\left| Out(V_j) \right|} \qquad (3.5)$$

where *d* is a parameter that is set between 0 and 1.

The score of each vertex is recalculated upon each iteration based on the new weights that the neighboring vertices have accumulated. The algorithm terminates when the convergence point is reached for all the vertices, meaning that the error rate for each vertex falls below a pre-defined threshold.

This vertex scoring scheme is based on a random-walk model, where a walker takes random steps on the graph, with the walk being modelled as a Markov process. Under certain conditions (the graph is aperiodic and irreducible), the model is guaranteed to converge to a stationary distribution of probabilities associated with the vertices in the graph. Intuitively, the stationary probability associated with a vertex represents the probability of finding the walker at that vertex during the random-walk, and thus it represents the importance of the vertex within the graph.

**HITS (Hyperlinked Induced Topic Search)** [Kle99] is an iterative algorithm that was designed for ranking Web pages according to their degree of "authority". The HITS algorithm makes a distinction between "authorities" (pages with a large number of incoming links) and "hubs" (pages with a large number of outgoing links). For each vertex, HITS produces two sets of scores – an "authority" score, and a "hub" score:

$$HITS_A(V_i) = \sum_{V_j \in In(V_i)} HITS_H(V_j) \qquad (3.6)$$

$$HITS_H(V_i) = \sum_{V_j \in In(V_i)} HITS_A(V_j) \qquad (3.7)$$

**PageRank** [Bri98] is perhaps one of the most popular ranking algorithms, and was designed as a method for Web link analysis. Unlike other ranking algorithms, PageRank integrates the impact of both incoming and outgoing links into one single model, and therefore it produces only one set of scores:

$$PR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{PR(V_j)}{|Out(V_j)|} \qquad (3.8)$$

where *d* is a parameter that can be set between 0 and 1. In matrix notation, the PageRank vector of stationary probabilities is the principal eigenvector for the matrix

*A$_{row}$*, which is obtained from the adjacency matrix *A* representing the graph, with all rows normalized to sum to 1: $P = A^T_{row}P$.

A ranking process starts by assigning arbitrary values to each node in the graph, followed by several iterations until convergence below a given threshold is achieved. Convergence is achieved when the error rate for any vertex in the graph falls below a given threshold, where the error rate of a vertex $V_i$ is approximated with the difference between the scores computed at two successive iterations: $S^{k+1}(V_i) – S^k(V_i)$ (usually after 25-35 iteration steps). After running the algorithm, a score is associated with each vertex, which represents the "importance" (rank) of the vertex within the graph. Note that for such iterative algorithms, the final value obtained for each vertex is not affected by the choice of the initial value, only the number of iterations to convergence may be different.

**Undirected Graphs:** Although traditionally applied on directed graphs, algorithms for node activation or ranking can be also applied to undirected graphs. In such graphs, convergence is usually achieved after a larger number of iterations, and the final ranking can differ significantly compared to the ranking obtained on directed graphs.

**Weighted Graphs:** When the graphs are built from natural language texts, they may include multiple or partial links between the units (vertices) that are extracted from text. It may be therefore useful to indicate and incorporate into the model the "strength" of the connection between two vertices $V_i$ and $V_j$ as a weight $w_{ij}$ added to the corresponding edge that connects the two vertices. Consequently, we introduce new formulae for graph-based ranking that take into account edge weights when computing the score associated with a vertex in the graph, e.g.

$$PR^W(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} w_{ij} \frac{PR^W(V_j)}{\sum_{V_k \in Out(V_j)} w_{jk}} \qquad (3.9)$$

## 3.6. Genetic Algorithms

A genetic algorithm (GA) uses the principles of evolution, natural selection, and genetics from natural biological systems in a computer algorithm to simulate evolution [Gol89]. Essentially, the genetic algorithm is an optimization technique that performs a parallel, stochastic, but directed search to evolve the fittest population.

Genetic algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. GAs are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. The more common applications of GAs are the solution of optimization problems, where efficient and reliable results have been shown. That is the reason why we will use these algorithms to find parameters for the rule base reduction methods.

In the early 1970s, John Holland introduced the concept of genetic algorithms. His aim was to make computers do what nature does. Holland was concerned with algorithms that manipulate strings of binary digits. Each artificial "chromosome" consists of 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 an ability to adapt and learn without being told what to do. In other words, nature finds good chromosomes blindly. GAs do the same. Two mechanisms link a GA to the problem it is solving: encoding and evaluation. The GA uses a measure of fitness of individual chromosomes to carry out reproduction. As reproduction takes place, the crossover operator exchanges parts of two single chromosomes, and the mutation operator changes the gene value in some randomly chosen location of the chromosome.

### 3.6.1. Basic genetic algorithm

The basic genetic algorithm consists of following steps [Neg02]:

Step1: Represent the problem variable domain as a chromosome of a fixed length; choose the size of a chromosome population $N$, the crossover probability $p_c$ and the mutation probability $p_m$.

Step2: Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basic for selecting chromosome that will be mated during reproduction.

Step3: Generate an initial population of chromosome of size $N$: $x_1$, $x_2$, ..., $x_N$.

Step4: Calculate the fitness of each individual chromosome:
$f(x_1)$, $f(x_2)$, ..., $f(x_N)$.

Step5: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness.

Step6: Create a pair of offspring chromosomes by applying the genetic operators— crossover and mutation.

Step7: Place the created offspring chromosomes in the new population.

Step8: Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population $N$.

Step9: Replace the initial (parent) chromosome population with the new (offspring) population.

Step10: Go to Step 4, and repeat the process until the termination criterion is satisfied.

GA represents an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500. The entire set of generations is called a run. Because GAs use a stochastic search method, the fitness of a population may remain stable for a number of generations before a superior chromosome appears. A common practice is to terminate a GA after a specified number of generations and then examine.

## 3.6.2. Representation, Population and Fitness Function

### Representation

Before applying a GA we first must encode the parameters of the problem to be optimized. GAs do not deal directly with the parameters, they work with codes that represent the parameters. Thus, the representation of the problem is the first important issue in the design of genetic algorithms, i.e., how to represent the problem parameters.

Different representation schemes might cause different performances in GAs [Cha99, Hau04, Mel99]. There are two common representation methods that we can use: *floating point* and *bit string*. The preferred method is the binary string because the majority of genetic operators are suitable for this type of representation, and also, this representation has a better impact in the performance of genetic algorithms. In binary representation of GAs each parameter to optimize is encoded using a binary string of a fixed length, so we need to find a codification function that maps a real parameter value into an integer in the interval $[0, 2^{l-1}]$ where $l$ is the length of the binary string. To construct such a function, we usually first decide the range of each parameter value based on background knowledge of the problem whose parameters we want to optimize. Based on the range and desired precision of the optimal value for each parameter we can calculate the length of the binary string required. The role of the codification function and its inverse (decodification function) is encoding and decoding a space of values (possible solutions) for a

parameter, such that we can pass from real parameter values to a binary string that can be used by GAs.

## Population

Genetic algorithms operate with a population of possible solutions, not only one, so, at the beginning a GA requires an initial population of individuals. The size of the initial population can be fixed, or depending of the algorithm, this can be adaptive. There are three ways of forming the initial population: *randomly*, *deterministic* and *by help of other methods*. The first methods generate solutions randomly. The second initializes the population with specified chromosomes, for instance, only chromosomes of 0's, 1's, and so on [Una05]. Also knowledge of the problem can be used and obtain solutions that satisfy certain requirements. Finally, the initial population can be also initialized with individuals proportioned by other optimization techniques.

## Fitness Evaluation Function

The fitness of an individual in genetic algorithms is the value returned by the fitness evaluation function. This evaluation function measures the fitness or quality of chromosomes to solve a problem. Obviously, the fitness of chromosomes less fit to solve a problem are more punished than the fitness of fitter chromosomes.

The fitness evaluation function acts as an interface between the genetic algorithm and the optimization problem. First, the chromosome must be decoded, and then evaluated by the fitness function which returns a value indicating the fitness of chromosomes to solve the problem. Fitness evaluation function plays an important role in GA because it provides information about how good a solution performs to solve the problem. This information guides the search of a genetic algorithm, and more accurately, the fitness evaluation function results to determine the likelihood that a possible solution is selected to produce new solutions in the next generation.

## *3.6.3. Genetic Operators*

### Crossover operator

Crossover is a genetic operator that combines two chromosomes (parents) to produce one o two chromosomes (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. First, the crossover operator randomly chooses a crossover point where two parent chromosomes "break", and then exchanges the chromosome parts after that point with a user-definable crossover probability. As a result, two new offspring are created. If a pair of chromosomes does not cross over, then the chromosome cloning takes place, and the offspring are created as exact copies of each parent [Neg02].

The most common forms of crossover are one-point, two-point, n-point, and uniform crossover showed in Figure 3.2.

### Mutation operator

Mutation represents a change in the gene (Figure 3.3). Its role is to provide and guarantee that the search algorithm is not trapped on a local optimum. The mutation operator flips a randomly selected gene in a chromosome. The mutation operator uses a mutation probability $p_m$ previously set by the user, which is quite small in nature, and it is kept low for GAs, typically in the range 0.001 and 0.01. According with this probability, the bit value is changed from 0 to 1 or vice versa. This way, an offspring is produced from a single parent [Neg02].
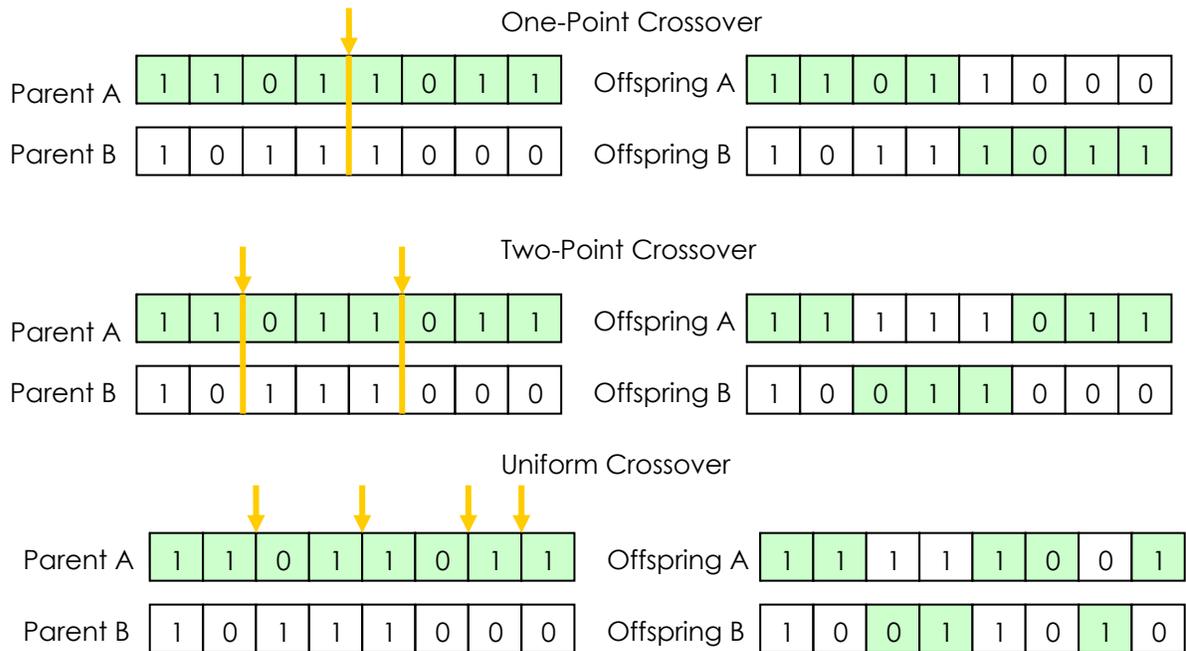
Figure 3.2 – Crossover operators.
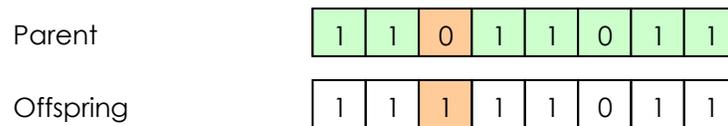


Figure 3.3 – Mutation operator.

The following three operators compose the *CHC* (Cross-generational elitist selection, Heterogeneous recombination by "incest prevention", and Cataclysmic mutation) algorithm which has the idea that recombination should be the dominant search operator.

**Elitist Selection and Incest Prevention**

After recombination, the *N* best unique individuals are drawn from the parent population and the offspring population to create the next generation. This also

implies that duplicate individuals are removed from population. This form of selection is also referred to as *truncation selection* [Esh91].

After the truncation selection, pairs of individuals are randomly formed with the new parent population to apply the recombination, forming *N/2* pairs of individuals. However, the CHC algorithm also employs a heterogeneous recombination restriction as a method of "*incest prevention*". This is accomplished by matting only those pairs of chromosomes which differ from each other by some number of bits, i.e. a *matting threshold*. The initial threshold is set at *L/4*, where *L* is the length of the string. If any of the *N/2* recombination could not be applied, i.e. if a generation occurs in which no offspring are inserted into the new children population, then the threshold is reduced by one. This means that the chromosomes of the population have become very similar.

**Half Uniform Crossover (HUX) operator**

In this operator bits are randomly and independently exchanged, but exactly half of the bits that differ between parents are swapped, see Figure 3.4. The HUX operator [Esh91] ensures that the offspring are equidistant between the two parents. This serves as a diversity preserving mechanism.
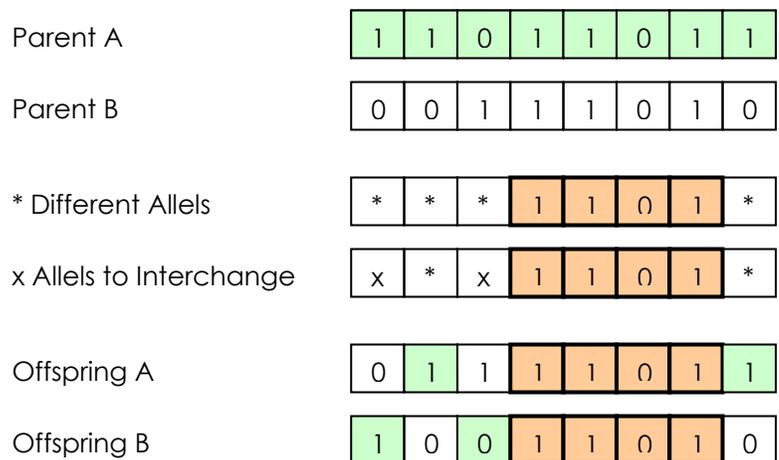


Figure 3.4 – Half Uniform Crossover.

**Cataclysmic Mutation**

No mutation is applied during the regular search phase of the CHC algorithm [Esh91]. When no offspring can be inserted into the population of a succeeding generation, and the mating threshold has reached a value of zero, CHC introduces new diversity into the population via a form of restart. *Cataclysmic mutation* uses the best individual in the population as a template to re-initialize the population. The new population includes a copy of the best individual; the remainder of the population is generated by applying a simple mutation relatively high, for instance 35%, of the best individual. The *new threshold* value will be the product of the chromosome's length (L) and the mutation's percentage (%) used to generate the new population. There are many other ways to refresh the population, for example, to rescue the best k individuals and generating the remainder randomly, or to rescue the best k individuals and use these as templates to generate the remaining of the population, and so on.

## 3.7. Clustering Algorithms

The clustering algorithms form groups of objects in order to archive the greatest possible similarity between objects of a group, and at the same time to keep the dissimilarity of the objects of other groups. More formally, the clustering problem [Jai99] is the problem of dividing a given set $\{x_1, ..., x_N\}$ of $N$ data points into several non-overlapping homogenous groups. Each such group or cluster should contain similar data items and data items from different groups should not be similar. We refer to a clustering in $k$ groups as a $k$-clustering.

Many different approaches to the clustering problem have been developed. Some operate on data represented by their coordinates in a feature space and others operate on a matrix of pairwise similarities between data points. To give a briefly overview of the different types of methods, we list them in three groups [Ver04]:

1. Hierarchical clustering methods. These produce a hierarchy of clusters for the data. The first level of the hierarchy contains all data and at each subsequent level of the hierarchy, one of the clusters of the previous level is split in two. The last level contains all data in individual clusters. The hierarchy is based on pairwise similarities between data points and is constructed either top-down or bottom-up.

2. Partitional clustering methods. These produce a single clustering with a fixed and (often) specified number of clusters. Most partitional clustering algorithms do not operate on the basis of pairwise similarities, but with data represented in some feature space. Typically, these methods start with an initial k-clustering and apply an iterative algorithm to improve upon the initial clustering according to some criterion. Most partitional clustering methods make, sometimes implicitly, assumptions on the distribution of data within each cluster. The partitional algorithms CURE, k-means [Har79] is used in this thesis.

3. Spectral clustering methods. These operate on a matrix with pairwise similarities between the data points. The optimal clustering is defined as the clustering that minimizes the 'normalized cut' criterion that depends on the size of the clusters and the total sum of the similarities between points that are assigned to different clusters. Unfortunately, finding the clustering that minimizes the normalized cut is an NP-complete problem. However, a relaxation of this optimization problem can be efficiently solved, and the solution is given by an eigenvector of the normalized similarity matrix. The solution of the relaxed problem is then further processed to find an approximate solution for the original problem. The term 'spectral clustering' refers to the normalized similarity matrix which can be used to assess the number of clusters in the data. Spectral methods are used both to find hierarchical clustering and *k*-clustering for a given *k*. We treat them separately since their working is quite different from the other approaches.

**Algorithm k-means**

The k-means algorithm [Har79] is an algorithm to cluster *n* objects based on attributes into *k* partitions, $k < n$. It is similar to the expectation-maximization algorithm for

mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. It assumes that the object attributes form a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or the squared error function

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2 \qquad (3.10)$$

where there are $k$ clusters $S_i$, $i$ = 1, 2, ..., $k$, and $\mu_i$ is the centroid or mean point of all the points $x_j \in S_i$.

# CHAPTER 4. Proposed Method for Single Text Summarization

The objective of this chapter is to present the proposed methods for generation of summaries for single document.

In the section 4.1 we present the first method for the generation of text summaries for a single document. Different term selection, term weighting, sentence weighting and sentence selection methods are described.

We propose another interesting way to generate summaries which is using the graph algorithms (section 4.2), which considerably improves the results for the generation of text summaries. Finally, some general ideas of genetic and clustering algorithm are proposed.

## 4.1. Definitions

### 4.1.1. Sequences of ngrams

An ngram is a sequence of *n* words. We say that an ngram occurs in a text if these words appear in the text in the same order immediately one after another. For example, a 4-gram (ngram of length 4) *words appear in the text occurs* once in the previous sentence, while *appear immediately after another* does not (these words do not appear on adjusting positions), neither does *the text appear in* (order is different).

The definition of ngram depends on what one considers words. For example, one can consider capitalized (*Mr. Smith*) and non-capitalized (*a smith*) words as the same word or as different words; one can consider words with the same morphological stem (*ask, asked, asking*), the same root (*derive, derivation*), or the same meaning (*occur, appear*) as the same word; one can omit the stop-words (*the, in*) when counting word positions, etc. Say, one can consider that in our example sentence above there occur the ngrams *we say* (capitalization ignored), *word appear* (plural ignored), *appear text* (*in the* ignored). This can affect counting the ngrams: if one considers occur and appear as equivalent and ignores the stop-words, then in our example sentence the bigram *appear text* occurs twice.

### 4.1.2. Frequent Sequences

We call an ngram frequent (more accurately, β-frequent) if it occurs more than β times in the text, where β is a predefined threshold. Frequent ngrams—we will also call them frequent sequences (FSs)—often bear important semantic meaning: they can be multiword expressions (named entities: *The United States of America*, idioms: *kick the basket*) or otherwise refer to some idea important for the text (*the President's speech*, *to protest against the war*).

In frequent sequential pattern mining, a sequence is extracted if it is repeated frequently in a collection of documents. For example, the following frequent sequences were obtained from 5 sentences (see Figure 4.1) with $\beta = 2$ showed in Figure 4.2.

*…El gobierno de Egipto protege las pirámides…*
*…Las pirámides de Egipto son un patrimonio cultural…*
*…Las pirámides fueron construidas por los faraones…*
*…Las pirámides de Egipto fueron tumbas para los faraones de Egipto...*
*…Un buen gobierno protege su patrimonio cultural...*

Figure 4.1 – Example of 5 sentences from an arbitrary text.

The discovery of frequent sequences as has been commonly used in sequential pattern mining, would not be very useful in analyzing texts because the number of frequent sequences that appear in a given text is very big. In Figure 4.2, frequent sequences are shown for $\beta = 2$. In this example, we obtained 19 frequent sequences.

1. *"gobierno"*
2. *"de"*
3. *"Egipto"*
4. *"protege"*
5. *"las"*
6. *"pirámides"*
7. *"un"*
8. *"patrimonio"*
9. *"cultural"*
10. *"fueron"*
11. *"los"*
12. *"faraones"*
13. *"de Egipto"*
14. *"las pirámides"*
15. *"pirámides de"*
16. *"patrimonio cultural"*
17. *"los faraones"*

18. *"las piramides de Egipto"*

19. *" patrimonio cultural"*

Figure 4.2 – Frequent sequences extracted from the example of Figure 4.1 with $\beta = 2$.

### *4.1.3. Maximal Frequent Sequences*

One way to reduce all frequent sequences is to take into account only those sequences that are not only frequent subsequences but in addition are maximal. FSs that are not parts of any other FS are called Maximal Frequent Sequences (MFSs) [Gar04, Gar06]. For the example of Figure 4.1, we obtained sequences but only 4 sequences are maximal. In this sense, one of the properties of MFSs is that all those subsequences that can be formed MFS are also frequent. In other words, this represents that MFS contains any frequent subsequence, thus bearing a compact representation.

1. *" gobierno"*
2. *" las piramides de Egipto"*
3. *" patrimonio cultural"*
4. *" los faraones"*

Figure 4.3 – MFSs for the example of Figure 4.1 with $\beta = 2$, GAP = 0.

Although MFSs "gobierno", "las piramides de Egipto", "patrimonio cultulral", "los faraones" (see Figure 4.3) have consistency, in practice it may not happen, because there is no restriction in the separation in the text between the words that form a sequence. The restriction in the separation is called GAP restriction. Thus, for small values of GAP we can find more understandable patterns and the context of words maintains. For example, in Figure 4.4, MFSs with a threshold $\beta = 3$ and a GAP equal to 0 are presented.

1. *"las piramides"*
2. *"de Egipto"*

Figure 4.4 – MFSs for the example from Figure 4.1 with $\beta = 3$, GAP = 0.

In the case of the discussed example, we obtain the same MFSs as in Figure 4.4 with $\beta = 4$ and GAP = 0. For example, with $\beta = 2$ and GAP = 2, we obtain the following MFSs showed in Figure 4.5.

1. *"gobierno protege"*

Figure 4.5 – MFSs for the example of Figure 4.1 with $\beta = 2$, GAP = 2.

The MFS from Figure 4.5 was obtained from sequences 1 and 4 in the way showed in Figure 4.6 with the words which are underlined. As we can see that there is a separation between words (two words as a given restriction).

*…El <u>gobierno</u> **de Egipto** <u>protege</u> las pirámides…*
*…Las pirámides de Egipto son un patrimonio cultural…*
*…Las pirámides fueron construidas por los faraones…*
*…Las pirámides de Egipto fueron tumbas para los faraones de Egipto...*
*…Un buen <u>gobierno protege</u> su patrimonio cultural...*

Figure 4.6 – Sequences utilized for obtaining MFS from Figure 4.5.

One of our hypotheses was that only MFSs should be considered as bearing important meaning, while non-maximal FSs (those that are parts of another FS) should not be considered. Our additional motivation was cost vs. benefit considerations: there are too many non-maximal FSs while their probability to bear important meaning is lower. In any case, MFSs represent all FSs in a compact way: all FSs can be obtained from all MFSs by bursting each MFS into a set of all its subsequences. García [Gar04] proposed an efficient algorithm to find all MFSs in a text, which we also used to efficiently obtain and store all FSs of the document.

The notions of FSs and MFSs are closely related to that of repeating bigrams; see Section 4.1.2. This set is conceptually simpler, but for computational implementation MFSs could be more compact.

### 4.1.4. Multiword Descriptions

Our hypothesis is that FSs can express ideas both important and specific for the document. This can be argued in terms of *tf-idf* (term frequency—inverse document frequency, a notion well-known in information retrieval [Bae99]): on the one hand, the idea expressed by an FS is important for the document if it repeatedly returns to it (high term frequency); on the other hand, the corresponding idea should be specific for this document, otherwise there would exist in the language a single word or at least an abbreviation to express it (high inverse document frequency).

An ngram can be a part of another, longer ngram. All ngrams contained in an FS are also FSs. However, with the arguments given above one can derive that such smaller ngrams may not bear any important meaning by their own: e.g., *The United States of America* is a compound named entity, while *The United* or *States of America* are not. Exceptions like *The United States* should not affect much our reasoning since they tend to be synonymous to the longer expression, and the author of the document would choose one or another way to refer to the entity, so they should not appear frequently both in the same document.

## 4.2. Proposed Method using Multiword Descriptions

### 4.2.1. Main Algorithm

We describe the general scheme of the proposed algorithm which consists of the four steps:

### 4.2.2. Term Selection

When we say the terms, we referred to the features which we use in this step. The terms are words, n-grams, or MFS's extracted from a document. The details of MFS's are described below. Also we extract terms derived from MFS's such as words and n-grams. Namely, we propose the following variants of term selection:

- $M$: the set of all MFSs, i.e., an ngram $m \in M$ if it is an MFS with some threshold $\beta$ (recall that MFSs are of 2 words or longer and $\beta \geq 2$).[2] In the example from Figure 4.7, $M$ = {*is the most beautiful, the most beautiful*}. Also, we denote by $M_2$ the set of all MFSs with $\beta = 2$.

- $B$: repeating bigrams, i.e., bigrams with frequency at least 2. It is easy to show that it is the same set as the set of all bigrams from MFSs: a bigram $b \in B$ iff there exists an MFS $m \in M$ such that $b \subseteq m$. What is more, considering in the latter definition $M_2$ instead of $M$ also gives the same set. In our example, $B$ = {*is the, the most, most beautiful*}.

- $W$: single words (unigrams) from elements of $B$ or, which is the same, of $M$. Namely, a word $w \in W$ if there exists a bigram $b \in B$ such that $w \in b$; it is easy to show that $w \in W$ iff there exists an MFS $m \in M$ such that $w \in m$. Again, considering $M_2$ in the latter definition also gives the same set. In our example, $B$ = {*is, the, most, beautiful*}.

- $N$: all ngrams from MFSs, i.e., an ngram $n \in N$ if there exists an MFS $m \in M$ such that $n \subseteq m$ (including single words, i.e., 1-grams). Again, considering in the latter definition $M_2$ also gives the same set, which allows for efficient calculation of the 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*}. Note that $W \subset N, M \subset N$.

- $N \setminus W, N \setminus M_2, N \setminus (W \cup M_2)$: same as $N$ but not including 1-grams, the whole 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*}.

    1. *… Mona Lisa is the most beautiful picture of Leonardo da Vinci …*
    2. *… Eiffel tower is the most beautiful tower …*
    3. *… St. Petersburg is the most beautiful city of Russia …*
    4. *… The most beautiful church is not located in Europe …*

Figure 4.7 – Example of 4 sentences from an arbitrary text.

---

[2]     In practice, we only considered the MFSs with the thresholds $\beta = 2$, 3, and 4, since MFSs with higher thresholds were very rare in our collection, except for those generated by stop-words.

We give different definitions of the sets *B* and *W* to show that they are naturally derived from the notion of MFS and at the same time can be efficiently calculated.

### 4.2.3. Term Weighting

We propose a scheme for weighting MFS which take into account $T_i$ frequency of MFS, length of MFS, and frequency of derived terms from MFS. The terms $T_i$ can be weighted in different manners, and have a weight $t_i$. This general scheme is defined as $p_i(t_j) = X \cdot Y$, where $p_i(t_j)$ - term weighting $j$ in the documents $i$, $X$ and $Y$ can be determined as frequency of MFS, length of MFS, and frequency of derived terms from MFS. This term weighting scheme permits to detect which of the characteristics of MFS helps better to summarize a text. Specifically, the following term weighting schemes are proposed:

– *f*: frequency of the term in MFSs, i.e., the number of times the term occurs in the text within some MFS. In our example, $f(is) = 3$ since it occurs 3 times in the text within the MFS is the most beautiful. If the term itself is an MFS, then this is just the frequency of this term in the text (e.g., for *M*, f is the same as term weight in Experiment 1; for *W* and *N* it is not). Under certain realistic conditions (MFSs do not intersect in the text, words do not repeat within one MFS) *f* is the number of times the term occurs in the text as part of a repeating bigram. In our example, $f(is) = 3$ since it occurs 3 times in a repeating bigram *is the* (and one time in a non-repeating context *church is not*).
– *l*: the maximum length of an MFS containing the term. In our example, $l(is) = 4$ since it is contained in a 4-word MFS *is the most beautiful*.
– 1: the same weight for all terms.

### 4.2.4. Sentence Weighting

For this stage, we calculate the sum of the weights of the terms contained in the sentence. When a sentence $S_i$ has weight $s_j = $ sum $w_{ij}$, contribution of $T_i$ in $S_j$ 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.

## 4.2.5. Sentence Selection

This procedure completes a summary adding the densest sentences or choosing the position of a sentence in a text until the summary is limited by the number of words. As the first option, we chose the sentences which have more weighting score. This type of methods is dominion independent and can be applied for a variety of texts. As the second option, the type of methods is position dependent and can be applied only for special topics. These two options are resumed as follows:

- best: sentences with greater weight were selected until the desired size of the summary (100 words) is reached. This is the most standard method.
- *k*best+first: *k* best sentences were selected, and then the first sentences of the text weight were selected until the desired size of the summary is reached. This was motivated by the very hard-to-beat baseline mentioned in Section 5.1: only the very best sentences according to our weighting scheme might prove to be above this baseline.

## 4.3. Proposed Method using Graph Algorithms

### 4.3.1. Term Selection
The main contribution of this method is the proposal of using multiword descriptions as nodes of a graph.
- Term selection:  *M, W*.

### 4.3.2. Term Weighting
- Term weighting: frequency of the term in MFSs (*f*); the maximum length of an MFS containing the term (*l*); the same weight for all terms (1).

### 4.3.3. Sentence Weighting
- Sentence weighting: using pagerank.

### 4.3.4. Sentence Selection

The sentences with greater weight were selected until the desired size of the summary is reached. We consider *best* and *kbest+first* options.

## 4.4. Proposed Method using Genetic Algorithms

### 4.4.1. Main Algorithm

In this section, we present the proposed method to find topline. The best result obtained for the given collection, we call topline. Our method can be applied to find topline not only for summarization corpus (for example, such as DUC-2001 till DUC-2007), but also for other tasks of natural language processing. The scheme of the proposed method is showed in Figure 3.1. Here we have the detailed algorithm of proposed genetic algorithm. The objective of the proposed genetic algorithm is not only to find topline, but also to find the best combination of the sentences.
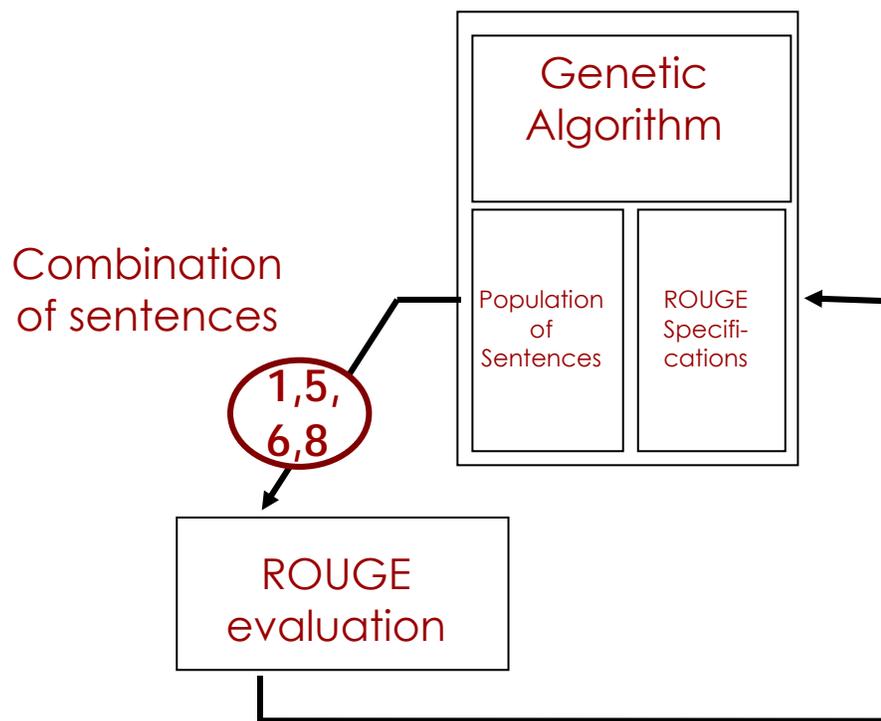
Figure 4.8 – Scheme of the proposed genetic algorithm.

The genetic algorithm maintains a population of chromosomes each of which represents a combination of candidate sentences. This genetic algorithm uses data from the system ROUGE to evaluate the fitness of each sentence in the population. It does this evaluation at each time step by simulating with each combination of the sentences and forming a fitness function based on the ROUGE evaluation which characterizes the desired performance. Using this fitness evaluation, the genetic algorithm propagates the number of sentences into the next generation via the combination of genetic operations proposed below. The combination of the sentences that is the fittest one in the population is used to compose a summary.

The proposed procedure of estimating the combination of sentences by GA is summarized as follows (see Figure 3.4):

1. Determine the number of sentences of the given text.
2. Construct an initial population.
3. Encode each chromosome in the population.
4. Evaluate the fitness value for each chromosome.
5. Reproduce chromosomes according to the fitness value calculated in Step 4.
6. Create offspring and replace parent chromosomes by the offspring through crossover and mutation.
7. Go to 3 until the maximum number of iterations is met.

**Representation**

To represent the combination of sentences, chromosomes of length $N \cdot B$ is used, where $N$ is calculated as the numbers of sentences of the original text and $B$ the number of bits which we use to encode the number of sentence.

**Population**

The initial population is formed randomly. Its size is fixed and equal to 35 individuals.

```
┌───────────────────────────────────────────────────────┐
│        Determine the combination of sentences          │
└───────────────────────────────────────────────────────┘
                            │
                            ▼
┌───────────────────────────────────────────────────────┐
│           Construct the initial population             │
└───────────────────────────────────────────────────────┘
                            │
                            ▼
┌───────────────────────────────────────────────────────┐
│        Encode each chromosome in the population        │
└───────────────────────────────────────────────────────┘
                            │
                            ▼
┌───────────────────────────────────────────────────────┐
│       Evaluate the fitness value of each chromosome    │
└───────────────────────────────────────────────────────┘
                            │
                            ▼
┌───────────────────────────────────────────────────────┐
│         Reproduce chromosomes according its fitness    │
└───────────────────────────────────────────────────────┘
                            │
                            ▼
┌───────────────────────────────────────────────────────┐
│     Create offspring and replace parent chromosomes by │
│      the offspring through crossover and mutation      │
└───────────────────────────────────────────────────────┘
       Number of generations
```
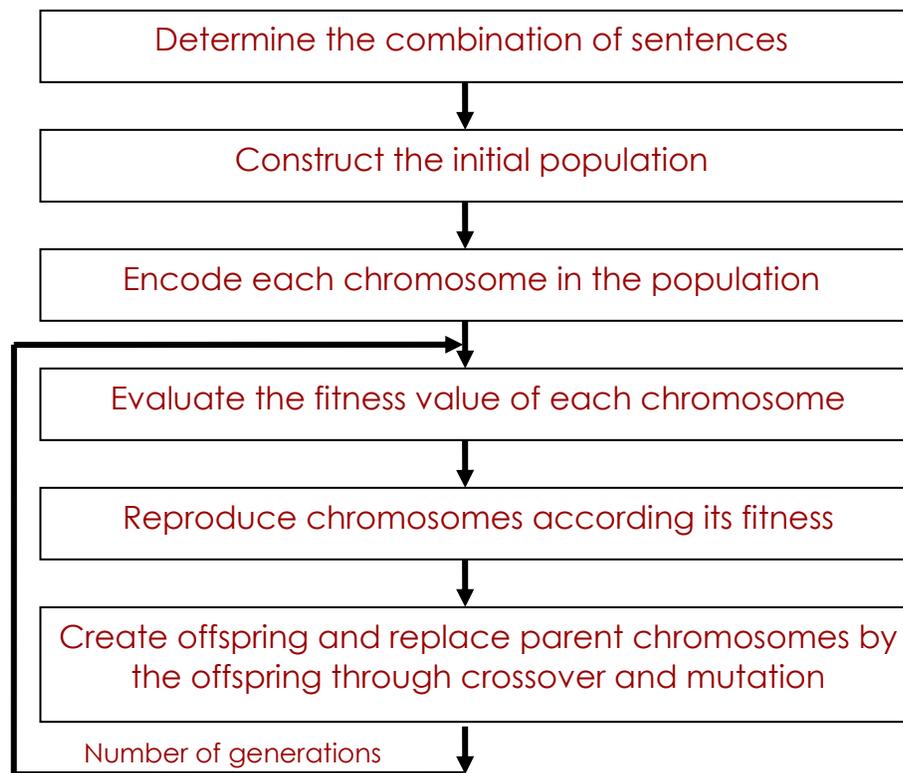
Figure 4.9 – Proposed genetic algorithm.

**Reproduction**

When the evaluation is done, we continue with the reproduction stage. In this step, some genetic operators were evaluated in an attempt to find the appropriate one. We consider in applying the strategy which more approximate to an equilibrium between diversity and convergence. Such strategy or algorithm (for example, CHC algorithm) involves the usage of HUX reproduction operator. So the new population is obtained by applying the HUX operator which ensures that offspring are equidistant between the two parents. This serves as a diversity preserving mechanism.

**Fitness Function**

We propose the fitness function so that it measures for each combination of sentences its F-measure using ROUGE evaluation system. And how close each combination of sentences in the population are met these specifications.

## 4.5. Proposed Method using Clustering Algorithms

### 4.5.1. Main Algorithm

In this thesis, we propose to apply a clustering algorithm with the objective to find more similar groups of sentences. From each group only one sentence will be chosen assuming that this sentence is the most representative. Thus, the resulted summary will be composed by the sentences extracted from each group.

For first experiments, we consider the clustering algorithm K-means [Har79].

1.  Stage of pre-processing: eliminate stopwords, apply stemming.

2.  Stage of clustering of the sentences:
    2.1.    Elaboration of vector model starting with sequences of the words of a document.
    2.2.    Determine the number of groups.
    2.3.    Assign initial seeds for clustering.
    2.4.    Generation of cluster of sentences.

3.  Stage of generation of sentences:
    3.1.    Search of the sentences more representative of each group.
    3.2.    Composition of the summary.

# CHAPTER 5. Experimental Results for Single Text Summarization

We test the proposed methods for composing document summaries on corpus DUC-2002. This is standard summarization collection in the English language using to compare the results of different text summarization methods.

In this chapter, various term selection, term weighting, sentence weighting, and sentence selection schemes are tested. They result from the proposed methods described in chapters 4 (single summarization methods). For each scheme the corresponding proposed method were applied and the composed summaries are evaluated. The obtained results are presented and discussed.

## 5.1. Experimental Settings

We have conducted several experiments to verify our hypotheses formulated in the previous chapters.

### 5.1.1. Algorithm

In each experiment, we followed the standard sequence of steps:

- Term selection: decide which features are to be used to describe the sentences;
- Term weighting: decide how the importance of each feature is to be calculated;
- Sentence weighting: decide how the importance of the features is to be combined into the importance measure of the sentence;
- Sentence selection: decide which sentences are selected for the summary.

The specific settings for each step varied between the experiments and are explained below for each experiment.

### 5.1.2. Test Data Set

We used the DUC-2002 collection provided [Duc]. In particular, we used the data set of 567 news articles of different length and with different topics. Each document in the DUC collection is supplied with a set of human-generated summaries provided by two different experts.[3] While each expert was asked to generate summaries of different length, we used only the 100-word variants.

### 5.1.3. Evaluation Procedure

---

[3]      While the experts were supposed to provide extractive summaries, we observed that the summaries provided in the collection were not strictly extractive: the experts considerably changed the sentences as compared with the original text.

We used the ROUGE evaluation toolkit [Lin03a]. This system is found to highly correlate with human judgments [Lin03b]. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses different statistics such as ngrams co-occurrences, longest common subsequence, weighted subsequence, skip-bigrams co-occurrence, etc. (see description in Section 2.1.2). Our evaluation was done using ngram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%.

### 5.1.4. Baseline

We denote *Baseline*: *first* the baseline, which selects the first sentences of the text until the desired size of the summary is reached [Duc]. The baseline configuration selects the first sentences of the text until the desired size is reached [Duc]. This configuration gives very good results on the kind of the texts (news reports) that we experimented with, but would not give so good results on other types of texts. Thus we proposed another baseline, denoted *Baseline*: *random*, which selects random sentences; the results presented below are averaged by 10 runs (for results see Table 5.4). We believe this to be a more realistic baseline for the types of texts other than news reports.

## 5.2. Experimental Methodology

We test the proposed methods using different proposed configurations of term selection, term weighting, sentence weighting, and sentence selection. We propose the following experimental methodology:

- Experiment 1: Different term selection options are tested.
- Experiment 2: Term selection using multiword description extracted for each sentence separately, term weighting and sentence selection.
- Experiment 3: Term selection using multiword description extracted for a collection of sentences (in other words, for a whole 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 using the phase of pre-processing.
- Experiment 6: Term selection, term weighting and sentence selection using DimaspCn.
- Experiment 7: Term selection, term weighting and sentence selection using graph algorithm.
- Experiment 8: Term selection, term weighting and sentence selection using genetic algorithm.
- Experiment 9: Term selection, term weighting and sentence selection using clustering algorithm.

## 5.3. Results

### 5.3.1. Term Selection

#### Experiment 1

For term selection, we compared MFSs with more traditional features such as single words and ngrams.

Optionally, stop-words were eliminated at the pre-processing stage; in this case our bigrams (or MFSs) could span more words in the original text, as explained in Chapter 3.

For term weighting, the frequency of the term was used; for sentence weighting, the sum of the weights of the terms contained in the sentence was used; for sentence selection, the sentences with greater weight were selected until the desired size of the summary (100 words) is reached.

#### Discussion

As a kind of statistical significance check, we randomly divided our test data into two halves and ran this (and most of the other) experiments separately on each subset. These experiments confirmed the qualitative observations reported in this paper.

As Table 5.1 shows, MFSs are a promising choice for term selection. This motivated our further experiments with term selection schemes derived from them, as well as with term weighting options for them.

Table 5.1 shows the results. Since the size of all summaries is the same (100 words), either measure (recall or precision) can be used for comparison.

Table 5.1 – Recall on 100-words summaries for different term selection options.

| Terms | With stop-words | Without stop-words |
|---|---|---|
| *W*: words from B or M | 0.39421 | 0.41371 |
| *B*: repeating bigrams | 0.40810 | 0.42173 |
| *M*: all MFSs | 0.43066 | 0.44085 |

## *5.3.2. Term selection (extracted for sentence), Term Weighting and Sentence Selection.*

### *Experiment 2*

Inspired by the above results, we further experimented with MFSs and other term selection options derived from them. In addition to *M*, we consider an option *W* from Section 3.

The results are shown in Table 5.2. We conducted our experiments in three phases. From Table 5.1 we knew that term selection scheme *M* with stop-words removed gave the best results with other parameters fixed (term weighting, sentence weighting, and sentence selection). So we started from modifying these parameters for the same term selection scheme; see the upper third part of Table 5.2. The first line of the table represents the best result from Table 5.1. The best results are highlighted in boldface.

In each experiment, we consider the following configuration of the main algorithm:

– Pre-processing: Optionally, stop-words were eliminated at the pre-processing stage.

– Term selection: Each original text is represented separately by each sentence. MFSs are extracted from each sentence separately. Resulted multiword descriptions extracted from each sentence are different from multiword descriptions extracted from a complete document. Specifically, the representation of a text is different, with the consequence that resulted patterns are different.

– Term weighting: frequency of the term in MFSs (*f*); the maximum length of an MFS containing the term (*l*); the same weight for all terms (1).

– Sentence weighting: the sum of the weights of the terms contained in the sentence was used.

– Sentence selection: the sentences with greater weight were selected until the desired size of the summary is reached *(best)*; *k* best sentences were selected, and then the first sentences of the text weight were selected until the desired size of the summary is reached *(kbest+first)*.

*Discussion*

Then we tried other term selection options, such as *W*, with the term weighting option 1 and the options related to *f*, which showed good performance in the first experiment. The results are shown in the middle third of Table 5.2. Term selection *W* gave better result than *M*. Finally, with the best combinations obtained from the first two experiments, we tried different sentence selection variants; see the last third of Table 5.2.

One can observe that any *k*best+first sentence selection option outperformed any combination that used the standard sentence selection scheme, with smaller *k* always giving better results—that is, only the slightest correction to the baseline improved it. The best result was obtained with single words derived from MFSs, with their weighting by the frequency of the corresponding MFS.

Table 5.2 – Results of the experiment where multiword descriptions are extracted from each sentence.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $l \times f$ | 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** |
| | $l \times l$ | | 0.42839 | 0.44633 | 0.43685 |
| | $l \times\times f$ | | 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 | <u>**0.46523**</u> | <u>**0.48219**</u> | <u>**0.47344**</u> |
| | | 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 |

## 5.3.3. Term selection (extracted for document), Term Weighting and Sentence Selection

### Experiment 3

In each experiment, we consider the following configuration of the main algorithm:

- Pre-processing: Optionally, stop-words were eliminated at the pre-processing stage.
- Term selection: Each original text is represented as a collection of sentences. MFSs are extracted from a complete document. In this experiment, in addition to *M* and *W* from experiment 2, we considered an option *N* and generalization of the sets $N$, $N \setminus W$, $N \setminus M_2$, $N \setminus (W \cup M_2)$.
- Term weighting: frequency of the term in MFSs (*f*); the maximum length of an MFS containing the term (*l*); the same weight for all terms (1).
- Sentence weighting: the sum of the weights of the terms contained in the sentence was used.
- Sentence selection: the sentences with greater weight were selected until the desired size of the summary is reached *(best)*; *k* best sentences were selected,

and then the first sentences of the text weight were selected until the desired size of the summary is reached *(kbest+first)*.

Table 5.3 – Results for different term selection options.

| Term Selection | | Term Weighting | Sentence Selection | Results | | |
|---|---|---|---|---|---|---|
| Terms | Stop-words | | | 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 |
| | excluded | $l \times f$ | | 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 | <u>**0.46576**</u> | <u>**0.48278**</u> | <u>**0.47399**</u> |
| | | | 2best+first | 0.46158 | 0.47682 | 0.46895 |
| M | excluded | 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 |

Then we tried other term selection options, such as *W* and *N*, with the term weighting option 1 and the options related to *f*, which showed good performance in the first experiment. The results are shown in the middle third of Table 5.3. Term selection *W* gave a slightly better result than *M*. The results for *N* are equal with *f* and 1 as weighting. Other combinations based on *N* did not give good results; see Table 5.4 (stop-words excluded, best sentence selection). Finally, we tried different sentence selection variants; see the last third of Table 5.3.

Table 5.4 – Results for variants of the 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 |
| $N \setminus W$ | 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*

One can observe that the results for experiment 2 are better than for experiment 3. Therefore, we consider for the comparison the results of experiments 1 and 3:

– State of the art: The author of [Mih04, Mih06] provided us with her data, which were evaluated in the same conditions as proposed methods. Specifically, *DirectedBackward* version of *TextRank* [Mih04] was evaluated. We also list the results of the original *TextRank* with implementation of *PageRank* with *DirectedBackward* version of *TextRank* but with some additional data processing to remove noisy data [Mih06] and the modified *TextRank* with a biased version of *PageRank* [Has07]. See details of the preprocessing in [Mih04, Mih06, Has07].
– Baseline: we use *Baseline*: *first* and *Baseline*: *random* (see Section 5.1).
– Our proposal: We compare these methods with the best results obtained with our proposal with the best and 1*best+first* sentence selection scheme, as shown in Table 5.3. In both cases our best results were obtained with the options *W* without stop-words for term selection and *f* for term weighting.

For fair comparison, we separated the methods by the type of information they used in addition to the weighting derived from terms:

– None (text is considered as a bag of sentences, sentence as a bag of terms, terms as strings),
– Order of sentences (say, first sentences are treated specially),

– Sophisticated pre-processing to obtain the terms.

We believe that in the future combination of these types of additional information can give even better results. The comparison is given in Table 5.5.

Table 5.5 – Comparison of results of experiment 3 with other methods.

| Additional info used | Method | Recall | Precision | F-measure |
|---|---|---|---|---|
| | Baseline: *random* | 0.37892 | 0.39816 | 0.38817 |
| None | TextRank: [Mih04] | **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, 1best+first* | **0.46576** | **0.48278** | **0.47399** |
| Pre-processing | TextRank: [Mih06] | 0.46582 | 0.48382 | 0.47450 |
| | TextRank: [Has07] | <u>**0.47207**</u> | <u>**0.48990**</u> | <u>**0.48068**</u> |

We could not apply our method with the pre-processing option because we did not have access to the specific details of the pre-processing procedure used in [Mih06] and [Has07] (see Experiment 5 for detail of pre-processing). However, in the other two categories our method outperformed the others. Possibly with the same type of pre-processing our method would outperform the others in the last category, too.

*Discussion*

We observed that words from repeating bigrams are good terms, and so are MFSs (we can speculate that MFSs are still better semantic units but splitting them into single words gives a more flexible and less sparse comparison). For term weighting, we observed that a good weighting scheme is the number of occurrences of the term in the text as part of a repeating bigram. With these settings, we obtained the results superior to the existing state-of-the-art methods.

Most of the state-of-the-art methods perform worse than the baseline method that takes into account a special ordering of sentences in news reports, which contain a nearly ready abstract in their first sentences. However, our methods can select one sentence better than this baseline method (while already the second-best sentence

selected by our method proves to be worse than the baseline). This gives a hybrid method (one sentence our and then back-off to the baseline) superior to both the baseline and other state-of-the-art methods.

In this experiment we did not apply pre-processing that was shown to be beneficial for other methods, so our results are below those of other methods when they do apply it, though above them when they do not. The latter makes us believe that when we apply pre-processing we will obtain results superior to all existing methods. This will be one of the experiments described below.

On the other hand, our experiments show that very different options (some of them rather absurd) only slightly affect the overall result, at least on the collection we used for our experiments. This can probably be explained by the nature of the texts in this collection (short news reports) and maybe by the behavior of the ROUGE evaluation scheme: the completely random selection baseline is rather high (so nearly any method would give at least similar results) while what seems to be almost top-line— selecting the first sentences of the text[4]—is quite low and quite near to the random baseline. This makes us rather pessimistic about much further progress in the results unless another data collection is used and probably better evaluation schemes are developed.

## 5.3.4. Term selection, Term Weighting and Sentence Selection with Different Thresholds

### Experiment 4

For this experiment, we use the configuration of the algorithm of experiment 3 (see Experiment 3). Then we tested that configuration with $\beta = 2$, 3, 4 (Table 5.3). We tested the proposed schemes with $\beta = 2$ (see Table 5.6), $\beta = 3$ (see Table 5.6), and $\beta = 4$ (see Table 5.8). The comparison results are shown below in Tables 5.9 – 5.11.

---

[4]         We believe this to be nearly top-line because the first lines of a news report are intended by its author to serve as a ready summary of the whole report, and are probably the best summary for it.

Table 5.6 – Results for experiment 4 with $\beta = 2$.

| Term Selection | | Term Weighting | Sentence Selection | Results | | |
|---|---|---|---|---|---|---|
| Terms | Stop-words | | | Recall | Precision | F-measure |
| M | excluded | $l \times f$ | 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 | | $f$ | 1best+first | <u>**0.46536**</u> | <u>**0.48230**</u> | <u>**0.47355**</u> |
| | excluded | | 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 5.7 – Results for experiment 4 with $\beta = 3$.

| Term Selection | | Term Weighting | Sentence Selection | Results | | |
|---|---|---|---|---|---|---|
| Terms | Stop-words | | | Recall | Precision | F-measure |
| M | excluded | $l \times f$ | best | 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 | <u>**0.46622**</u> | <u>**0.48407**</u> | <u>**0.47486**</u> |
| | | | 2best+first | 0.46223 | 0.47806 | 0.46989 |
| | | 1 | 1best+first | 0.45674 | 0.47551 | 0.46582 |
| M | | $l$ | 1best+first | 0.46631 | 0.48392 | 0.47483 |
| | | | 2best+first | 0.46007 | 0.47638 | 0.46796 |

Table 5.8 – Results for experiment 5 with $\beta = 4$.

| Term Selection | | Term Weighting | Sentence Selection | Results | | |
|---|---|---|---|---|---|---|
| Terms | Stop-words | | | Recall | Precision | F-measure |
| M | excluded | $l \times f$ | best | 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 | <u>**0.46788**</u> | <u>**0.48537**</u> | <u>**0.47634**</u> |
| | | | 2best+first | 0.46397 | 0.47985 | 0.47165 |
| | | 1 | 1best+first | 0.45674 | 0.47551 | 0.46582 |
| M | | $l$ | 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 proposed schemes using different thresholds. Here, we compare best results of the actual experiment (see Tables 5.9 – 5.11). We detect that the best configuration for MFSs as selected terms was obtained with combination of threshold ($β$ = 2, 3, 4). Also, we detect that for the terms derived from MFSs, the best threshold is $β$ = 2. The results of the configuration with combination of sentences with $β$ = 4 is the best obtained result.

Table 5.9 – Comparison of results using different thresholds (terms are MFS).

| Method | Recall | Precision | 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 5.10 – Comparison of results using different thresholds (terms derived from MFS).

| Method | Recall | Precision | 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 5.11 – Comparison of results using different thresholds (combination of sentences).

| 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 [Mih06] than baseline with little differences of the results. In previous experiment, we obtained better results than baseline. For the

collection of DUC2002 the results of baseline configuration is very high because the majority of texts is consisted of news descriptions and in such type of texts is common that the first sentences describe briefly the given news. In other words, some of the first sentences are abstract or summary of a given file. In other types of texts the configuration of baseline will not work at all. So, it is fair to compare with the state-of-the-art methods like Random Walks [Mih06]. The author of this work provided the data of its summaries which were evaluated in the same conditions as proposed methods. Specifically, *DirectedBackward* version of *TextRank* was evaluated (see Table 12, *TextRank*). Finally, the best of the proposed methods is included.

Table 5.12 – Comparison of results of experiments 2 and 3 with other methods.

| Additional info used | Method | Recall | Precision | F-measure |
|---|---|---|---|---|
| None | Baseline: *random* | 0.37892 | 0.39816 | 0.38817 |
|  | TextRank: [Mih04] | **0.45220** | 0.43487 | 0.44320 |
|  | **Proposed**: *Z, best* | 0.44659 | **0.45968** | **0.45293** |
| Order of sentences | Baseline: *first* | 0.46407 | 0.48240 | 0.47294 |
|  | **Proposed**: *Z*, 1*best+first* | **0.46788** | **0.48537** | **0.47634** |

We tested new method for the automatic generation of text summaries for a single document based on the discovery of MFSs, specifically we tested different combinations of term selection, term weighting, sentence weighting and sentence selection schemes with different thresholds. With first experiment, we observed that MFSs are good terms and help us to obtain good results comparing with words and n-grams. In the second experiment, we tested the proposed schemes with different thresholds. We conclude that words derived from MFSs are the best terms with $\beta = 2$ and MFSs are good terms with $\beta = 2, 3, 4$.

## 5.3.5. Pre-processing

### Experiment 5
The results of experiment are presented in Table 5.13. The best results are highlighted with bold type of letter. We detect that the weighting scheme of frequency of words derived from MFSs gives the best sentence for a summary, and together with sentences obtained with baseline configuration, the best summary is obtained. For the first part of this experiment, we extract MFSs excluding stopwords.

Table 5.13 – Results for configuration of experiment 2 using pre-processing (stop-words excluded).

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | 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 | <u>**0.46266**</u> | <u>**0.47979**</u> | <u>**0.47094**</u> |
| | *f* | 1best+first | 0.44605 | 0.44771 | 0.44676 |

For the second part of this experiment, we change pre-processing configuration: MFS are stemmed and stopwords are excluded from MFSs. See results in Table 5.14.

Table 5.14 – Results for configuration of experiment 2 using pre-processing (stemming and stopwords excluded).

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | 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 | <u>**0.46456**</u> | <u>**0.48169**</u> | <u>**0.47285**</u> |
| | *f* | 1best+first | 0.46432 | 0.48139 | 0.47258 |

For the third part of this experiment, MFS are stemmed and stopwords are included. The results are shown in Table 5.15.

Table 5.15 – Results for configuration of experiment 2 using pre-processing (stemming but stopwords kept).

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $l \times f$ | 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 | <u>**0.46508**</u> | <u>**0.48233**</u> | <u>**0.47343**</u> |

*Comparison with experiment 3*

We compare with the state-of-the-art methods like TextRank [Mih06]. Specifically, DirectedBackward version of TextRank was evaluated in the same conditions as proposed methods (see Table 5.16, TextRank) and the same version of TextRank with pre-processing (see Table 5.16, TextRank w/pre-processing). And also we compare results presented in Experiment 3 (see Table 5.16, MFS w/o pre-processing). Finally, the best version of each experiment is included (see MFS w/ pre-processing 1, 2, and 3).

We can see that pre-processing does not affect positively obtaining terms for extractive summarization, at least not in the case of MFSs.

*Discussion*

We modified our automatic single document text summarization method based on MFSs as terms by including pre-processing stage. We found, however, that pre-

processing does not affect positively the summaries obtained with our method. This is good news and bad news. Bad because we did not find better terms, and our summaries did not improve. Good because we confirmed that classic plain MFSs (sequences of wordforms and not stems or only significant words), which are calculated in a totally language-independent manner, are good terms for this task.

Table 5.16 – Comparison of results of pre-processing with other methods.

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| TextRank | 0.45220 | 0.43487 | 0.44320 |
| TextRank w/pre-processing | 0.46582 | 0.48382 | 0.47450 |
| MFS w/o pre-processing | **0.46576** | **0.48278** | **0.47399** |
| MFS w/pre-processing 1 | 0.46266 | 0.47979 | 0.47094 |
| MFS w/pre-processing 2 | 0.46456 | 0.48169 | 0.47285 |
| MFS w/pre-processing 3 | 0.46508 | 0.48233 | 0.47343 |

On the other hand, since we showed that our pre-processing almost does not either affect the results negatively, one can exclude stopwords and word endings from processing and still obtain almost the same quality of extractive summarization. Excluding stop-words significantly reduces the risk of exponential explosion of the size of the data structures used to mine for MFSs and for their application in our method, as well as the number of the terms (MFSs or ngrams) dealt with.

## 5.3.6. Graph Algorithm

*Experiment 6*

In each experiment, we consider the following configuration of the graph algorithm:

- Term selection: *M, W.*
- Term weighting: frequency of the term in MFSs (*f*); the maximum length of an MFS containing the term (*l*); the same weight for all terms (1).
- Sentence weighting: using pagerank.
- Sentence selection: the sentences with greater weight were selected until the desired size of the summary is reached.

Table 5.17 – Results of the graph algorithm (Experiment 6) using algorithm 1.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $f$ | best | 0.48009 | 0.47757 | 0.47865 |
| | $f^2$ | | 0.48056 | 0.47801 | 0.47910 |
| | $l$ | | 0.46668 | 0.48337 | 0.47474 |
| | $l$ | | 0.48025 | 0.47773 | 0.47881 |
| | $l^2$ | | **0.48058** | **0.47812** | **0.47917** |
| | $f \times l$ | | 0.48060 | 0.47810 | 0.47916 |
| | $f \times\times l$ | | 0.48079 | 0.47831 | 0.47937 |
| W | $f$ | best | <u>**0.48659**</u> | <u>**0.48324**</u> | <u>**0.48473**</u> |
| | $l$ | | 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** |

Table 5.18 – Results of the graph algorithm (Experiment 6) using algorithm 2.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | 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 |
| | $f \times l$ | | 0.48796 | 0.48529 | 0.48620 |
| | $f \times\times l$ | | 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** |

## 5.3.7. Genetic Algorithm

*Experiment 7*

First, we order all documents based on the number of sentences. First documents in the list are short document with small number of sentences, and in the end of list, the documents are long with a lot of sentences. We calculate topline trying all combinations of sentences in one document. The best combination of sentences, which has the highest score of F-measure, is chosen as topline result for a given document.

The average result of topline is 0.62971 (for 350 documents), see Table 5.19. It is not possible to find the best combination of sentences for all documents because of dimensionality explosion for long documents. That is why, we propose to use genetic algorithm to find the result of topline. The average result of topline using GA is 0.5931 (for 568 documents), see Table 5.20. The final topline result is 0.5960 (see Table 5.21).

Table 5.19 – Topline results trying all combination of sentences.

| 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 |
| between 300-400 | 0.60715 |
| **total** | **0.62971** |

Table 5.20 – Topline results using 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** |

Table 5.21 – Final topline results considering all combination of sentences (0-299) and proposed GA (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** |

## 5.3.8. Clustering Algorithm

### Experiment 8

In each experiment (see Table 5.22-5.25), we consider the following configuration of the proposed algorithm:

- Preprocessing: eliminate stop-words, then apply Porter stemming [Por80];
- Term selection: decide which size of n-grams as features are to be used to describe the sentences;
- Term weighting: decide how the importance of each feature is to be calculated, it can be BOOL, TF, IDF or TFIDF;
- Sentence clustering: decide the initial seeds for the K-means algorithm, in this case Baseline sentences;
- Sentence selection: after K-means finishes, select the closest sentence (most representative sentence) to each centroid for composing the summary;

Table 5.22 – Recall for different sizes of n-grams and its weights.

| Term Selection | Term Weighting | | | |
|---|---|---|---|---|
| | BOOL | TF | IDF | TFIDF |
| 1-grams | 0.47517 | 0.47686 | 0.47632 | 0.47545 |
| 2-grams | 0.47705 | 0.47694 | 0.47779 | 0.47777 |
| 3-grams | 0.47940 | 0.47940 | 0.47932 | 0.47932 |
| 4-grams | 0.47891 | 0.47891 | 0.47916 | 0.47913 |
| 5-grams | 0.47942 | 0.47942 | 0.47910 | 0.47910 |
| 6-grams | 0.47989 | 0.47979 | 0.48020 | 0.48020 |
| 7-grams | 0.47976 | 0.47992 | 0.47964 | 0.47993 |
| 8-grams | **0.48113** | 0.48072 | **0.48075** | 0.48055 |
| 9-grams | **0.48084** | **0.48084** | 0.48020 | **0.48109** |
| 10-grams | 0.48058 | **0.48103** | **0.48155** | **0.48101** |
| 11-grams | 0.48004 | 0.47903 | 0.47856 | 0.47856 |

Table 5.23 – Precision for different sizes of n-grams and its weights.

| Term Selection | Term Weighting | | | |
|---|---|---|---|---|
| | BOOL | TF | IDF | TFIDF |
| 1-grams | 0.47039 | 0.47219 | 0.47168 | 0.47078 |
| 2-grams | 0.47211 | 0.47204 | 0.47284 | 0.47284 |
| 3-grams | 0.47452 | 0.47454 | 0.47441 | 0.47441 |
| 4-grams | 0.47410 | 0.47410 | 0.47432 | 0.47429 |
| 5-grams | 0.47462 | 0.47462 | 0.47432 | 0.47432 |
| 6-grams | 0.47495 | 0.47497 | 0.47530 | 0.47530 |
| 7-grams | 0.47493 | 0.47510 | 0.47487 | 0.47512 |
| 8-grams | 0.47633 | 0.47606 | **0.47588** | 0.47587 |
| 9-grams | 0.47632 | **_0.47632_** | 0.47553 | **_0.47654_** |
| 10-grams | 0.47575 | **0.47609** | **_0.47684_** | **0.47634** |
| 11-grams | 0.47529 | 0.47409 | 0.47370 | 0.47370 |

Table 5.24 – F-measure for different sizes of n-grams and its weights.

| Term Selection | Term Weighting | | | |
|---|---|---|---|---|
| | BOOL | TF | IDF | TFIDF |
| 1-grams | 0.47264 | 0.47439 | 0.47387 | 0.47298 |
| 2-grams | 0.47445 | 0.47436 | 0.47519 | 0.47517 |
| 3-grams | 0.47683 | 0.47684 | 0.47673 | 0.47673 |
| 4-grams | 0.47638 | 0.47638 | 0.47661 | 0.47658 |
| 5-grams | 0.47689 | 0.47689 | 0.47658 | 0.47658 |
| 6-grams | 0.47729 | 0.47725 | 0.47762 | 0.47762 |
| 7-grams | 0.47721 | 0.47738 | 0.47713 | 0.47739 |
| 8-grams | **_0.47860_** | 0.47826 | **0.47818** | 0.47808 |
| 9-grams | **0.47845** | **_0.47845_** | 0.47773 | **_0.47868_** |
| 10-grams | 0.47803 | **0.47842** | **_0.47906_** | **0.47854** |
| 11-grams | 0.47753 | 0.47642 | 0.47599 | 0.47599 |

Table 5.25 – F-measure for different configurations of the proposed clustering algorithm.

| Model | | Weighting | | | |
|---|---|---|---|---|---|
| | | BOOL | TF | IDF | TFIDF |
| Initial Baseline Centroids for K-means | Bag of words | 0.47264 | 0.47439 | 0.47387 | 0.47298 |
| | ngrams with $n = 2$ (bigrams) | 0.47445 | 0.47436 | 0.47519 | 0.47517 |
| | ngrams with $n = 3$ (trigrams) | **0.47683** | <u>**0.47684**</u> | 0.47673 | 0.47673 |
| | ngrams with $n = 4$ (tetragrams) | 0.47638 | 0.47638 | 0.47661 | 0.47658 |
| | ngrams with $n = 5$ (pentagrams) | <u>**0.47689**</u> | <u>**0.47689**</u> | 0.47658 | 0.47658 |
| | MFSs β = 2, Gap = 0 | 0.47022 | 0.46862 | 0.47050 | 0.46985 |
| Initial Aleatory Centroids for K-means | Bag of words | 0.44374 | 0.44037 | 0.43949 | 0.43996 |
| | ngrams with $n = 2$ (bigrams) | 0.43814 | 0.43824 | 0.43927 | 0.43953 |
| | ngrams with $n = 3$ (trigrams) | 0.44054 | 0.43519 | <u>**0.44644**</u> | 0.44127 |
| | ngrams with $n = 4$ (tetragrams) | 0.43511 | 0.43510 | 0.44027 | 0.43142 |
| | ngrams with $n = 5$ (pentagrams) | 0.43617 | 0.43892 | **0.44343** | 0.43604 |
| | MFSs β = 2, Gap = 0 | <u>**0.44522**</u> | 0.43979 | 0.43953 | 0.44056 |

*Discussion*

In this experiment, we tested an extractive automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. In particular, the K-means algorithm for creating groups of similar sentences was used. Then, from the groups of sentences, the most representative sentence was selected for composing the summary. Normally, the definition of the number of groups to form and the initial seeds of the groups are considered as disadvantages of K-means. However, these parameters are used to take advantage of Baseline sentences in order to improve the quality of the summaries. The proposed approach, in contrast to supervised

methods, does not need large amount of golden samples for training. Therefore, our proposed approach is more independent from language and dominion.

According to experimental results we demonstrate that the proposed approach obtains more favorable results than others state-of-the-art approaches; ranking our proposed approach in second place, very close to the first place. In addition, our proposed approach outperforms the Baseline (first) heuristic for F-measure results, except for 1-gram and BOOL weighting.

# CHAPTER 6. *Proposed Method and Results for Multi Text Summarization*

The objective of this chapter is to present the proposed methods for generation of summaries for a collection of documents.

In the section 6.1 we present the first method for the generation of text summaries for a collection of documents. This method is composed by the following steps:

1. Term Selection
2. Term Weighting
3. Sentence Weighting
4. Sentence Selection.

We propose another interesting way to generate summaries using graph, genetic algorithms, which we will report later.

## 6.1. Proposed Method using Multiword Descriptions

Summarizing a single text is difficult enough. But summarizing a collection of thematically related documents poses several additional challenges. In order to avoid repetitions, one has to identify and locate thematic overlaps. One also has to decide what to include of the remainder, to deal with potential inconsistencies between document, and, when necessary, to arrange events from various sources along a single timeline. For these reason, multi document summarization is less developed than single document summarization.

### 6.1.1. Description of the Proposed Method

For comprehension of this chapter, it is necessary to familiarize with the chapter 4 where new ideas and notation are introduced in details. Here, we describe briefly some ideas that we plan to try for multi summarization.

### 6.1.2. Term Selection

We consider the following variants of term selection:

- *M*: the set of all MFSs
- *W*: single words (unigrams) from elements of *M*.

### 6.1.3. Term Weighting

We consider frequency of the term in MFSs *(f)*: the maximum length of an MFS containing the term *(l)*; the same weight for all terms (1).

### 6.1.4. Sentence Weighting

We test the option calculating the sum of the weights of the terms contained in the sentence.

### 6.1.5. Sentence Selection

For the moment, we test only the option when the sentences with greater weight are selected until the desired size of the summary (100 words) is reached. The baseline of multi summarization is low comparing to the baseline of single summarization. So, it is no sense to test *kbest+first* option.

## 6.2. Experimental Methodology

We proposed the following experimental methodology for testing new methods for multi summarization. In the section below, we show only two proposed method using multiword descriptions. The rest of the experiments are already implemented. We are obtaining the results of these experiments and analysing them.

- Experiment 1: Term selection using multiword description extracted for a collection of documents, term weighting and sentence selection.
- Experiment 2: Term selection, term weighting and sentence selection using different thresholds.
- Experiment 3: Term selection, term weighting and sentence selection using the phase of pre-processing.
- Experiment 4: Term selection, term weighting and sentence selection using graph algorithm.
- Experiment 5: Term selection, term weighting and sentence selection using genetic algorithm.

## 6.3. Results

### 6.3.1. Term selection and Term Weighting

Results for the methods described in Section 6.1.1 are shown in Table 6.1. The experiments are realized with $\beta = 2, 3, 4$.

Table 6.1 – Results for different term selection, term weighting options.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $f$ | best | 0.31372 | 0.31986 | 0.31660 |
| | $f^2$ | | 0.31162 | 0.31870 | 0.31499 |
| | $1$ | | 0.30620 | 0.31347 | 0.30965 |
| | $l$ | | **0.31411** | **0.32199** | **0.31786** |
| | $l^2$ | | 0.29184 | 0.30275 | 0.29706 |
| | $f \times l$ | | 0.31329 | 0.32103 | 0.31696 |
| | $f \times\times l$ | | 0.28328 | 0.29592 | 0.28933 |
| W | $f$ | best | **0.31919** | **0.32494** | **0.32192** |
| | $1$ | | 0.26413 | 0.27828 | 0.27072 |
| | $f^2$ | | 0.30056 | 0.30764 | 0.30391 |

## 6.3.2. Term Selection and Term Weighting with Different Thresholds

For this experiment, we use the configuration of the algorithm of experiment 1 of this section. Then we tested the algorithm with $\beta = 2$ (see Table 6.2), $\beta = 3$ (see Table 6.3), and $\beta = 4$ (see Table 6.4). Comparison of the results for the proposed method is shown in Tables 6.5 – 6.6. The results of the state-of-the-art methods [Duc] for multi summarization are shown in Table 6.7.

Table 6.2 – Results for different term selection and term weighting options with $\beta = 2$.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $f$ | best | 0.29038 | 0.29749 | 0.29373 |
| | $f^2$ | | 0.29038 | 0.29749 | 0.29373 |
| | 1 | | 0.29038 | 0.29749 | 0.29373 |
| | $l$ | | **0.29881** | **0.30714** | **0.30279** |
| | $l^2$ | | 0.28837 | 0.29912 | 0.29351 |
| | $f \times l$ | | **0.29881** | **0.30714** | **0.30279** |
| | $f \times\times l$ | | 0.28455 | 0.29606 | 0.29006 |
| W | $f$ | best | **0.32238** | **0.32902** | **0.32557** |
| | 1 | | 0.26413 | 0.27828 | 0.27072 |
| | $f^2$ | | 0.29559 | 0.30061 | 0.29796 |

Table 6.3 – Results for different term selection and term weighting options with $\beta = 3$.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $f$ | best | 0.30601 | 0.31302 | 0.30933 |
| | $f^2$ | | 0.30601 | 0.31302 | 0.30933 |
| | 1 | | 0.30601 | 0.31302 | 0.30933 |
| | $l$ | | **0.31574** | **0.32286** | **0.31911** |
| | $l^2$ | | 0.29904 | 0.30852 | 0.30356 |
| | $f \times l$ | | **0.31574** | **0.32286** | **0.31911** |
| | $f \times\times l$ | | 0.28309 | 0.29270 | 0.28768 |
| W | $f$ | best | **0.32279** | **0.32826** | **0.32538** |
| | 1 | | 0.26413 | 0.27828 | 0.27072 |
| | $f^2$ | | 0.30934 | 0.31602 | 0.31253 |

Table 6.4 – Results for different term selection and term weighting options with $\beta = 4$.

| Term Selection | Term | Sentence | Results | | |
|---|---|---|---|---|---|
| Terms | Weighting | Selection | Recall | Precision | F-measure |
| M | $f$ | best | 0.30964 | 0.31648 | 0.31288 |
| | $f^2$ | | 0.30964 | 0.31648 | 0.31288 |
| | 1 | | 0.30964 | 0.31648 | 0.31288 |
| | $l$ | | **0.32326** | **0.32980** | **0.32636** |
| | $l^2$ | | 0.30825 | 0.31812 | 0.31296 |
| | $f \times l$ | | **0.32326** | **0.32980** | **0.32636** |
| | $f \times\times l$ | | 0.29651 | 0.30533 | 0.30079 |
| W | $f$ | best | **0.31855** | **0.32329** | **0.32076** |
| | 1 | | 0.26413 | 0.27828 | 0.27072 |
| | $f^2$ | | 0.30934 | 0.31602 | 0.31253 |

Table 6.5 – Comparison of results using different thresholds (terms are MFS).

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| M where $\beta = 2, 3, 4$ | 0.31411 | 0.32199 | 0.31786 |
| M where $\beta = 2$ | 0.29881 | 0.30714 | 0.30279 |
| M where $\beta = 3$ | 0.31574 | 0.32286 | 0.31911 |
| M where $\beta = 4$ | **0.32326** | **0.32980** | **0.32636** |

Table 6.6 – Comparison of results using different thresholds (terms derived from MFS).

| Method | Recall | Precision | F-measure |
|---|---|---|---|
| M where $\beta = 2, 3, 4$ | 0.31919 | 0.32494 | 0.32192 |
| M where $\beta = 2$ | **0.32238** | **0.32902** | **0.32557** |
| M where $\beta = 3$ | 0.32279 | 0.32826 | 0.32538 |
| M where $\beta = 4$ | 0.31855 | 0.32329 | 0.32076 |

Table 6.7 – Preliminary comparison of results for multi summarization.

| Method | Recall |
|---|---|
| 1st best method | 0.3578 |
| 2nd best method | 0.3447 |
| **Proposed** | 0.3264 |
| 3rd best method | 0.3264 |
| 4th best method | 0.3056 |
| 5th best method | 0.3047 |
| baseline | 0.2932 |

### 6.3.3. Discussion

We tested new method for the automatic generation of text summaries for a multi document based on the discovery of MFSs, specifically we tested different combinations of term selection, term weighting, sentence weighting and sentence selection schemes with different thresholds. Comparing to other methods, we did not receive the best results but considering that the proposed method is language and dominion independent, we think that the results are very encouraging.

We observed that MFSs are good descriptors if we take greater threshold, and words that derived from MFSs are good descriptors if we take smaller threshold. Then we noted that for term selection option *M,* the term weighting option *I* showed best performance in the all experiment. We plan to test other configuration of the proposed methods in order to improve our results.

# CHAPTER 7. Conclusions

In this chapter, the preliminary conclusions and plan of future work to conclude this thesis are presented.

## 7.1. Conclusions

In this dissertation, new methods were developed for single and multi document summarization. The development of these methods for automatic generation of text summarization allows us to contribute in an efficient way to the area of natural language processing.

Each of the proposed methods includes the description of term selection, term weighting, sentence weighting, and sentence selection steps. For each experiment, the configuration of the proposed methods was presented and the corresponding results were given. Also, the discussion of the experimental results and comparison between different experiments of this thesis and the state-of-the-art were explicitly specified.

We have reached most of the promised contributions. Particularly, the following essential objectives were reached:

✓ New methods for single document summarization and multi document summarization.

✓ New approaches to deal with the task of generation of summaries in a language and dominion independent way.

✓ New methods for generating text summarizes based on the discovery of multiword descriptions.

✓ Methods for automatic generation of text summaries which are superior to the state-of-the-art methods.

The proposed method was tested on corpus DUC-2002. This is standard summarization collection in the English language proposed in text summarization conference which facilitates the comparison of the results obtained by researchers of the area of text summarization.

Replying to the research question of this thesis, we can say that the most important parts of the text can be automatically detected using multiword descriptions proposed in this thesis. In particularly, we use maximal frequent sequences in order to find multiword descriptions.

The comparison results are shown in Figure 7.1. Topline was calculated in order to find the result of evaluation for the best summary. In the same manner, the result of evaluation for the worst summary can be obtained used the proposed method in this thesis. However, random selection of sentences for composing a summary is considered as the worst summary. Also, considering that F-measure do not adequately qualify summaries, we can recalculate the results as shown in Figure 7.1.
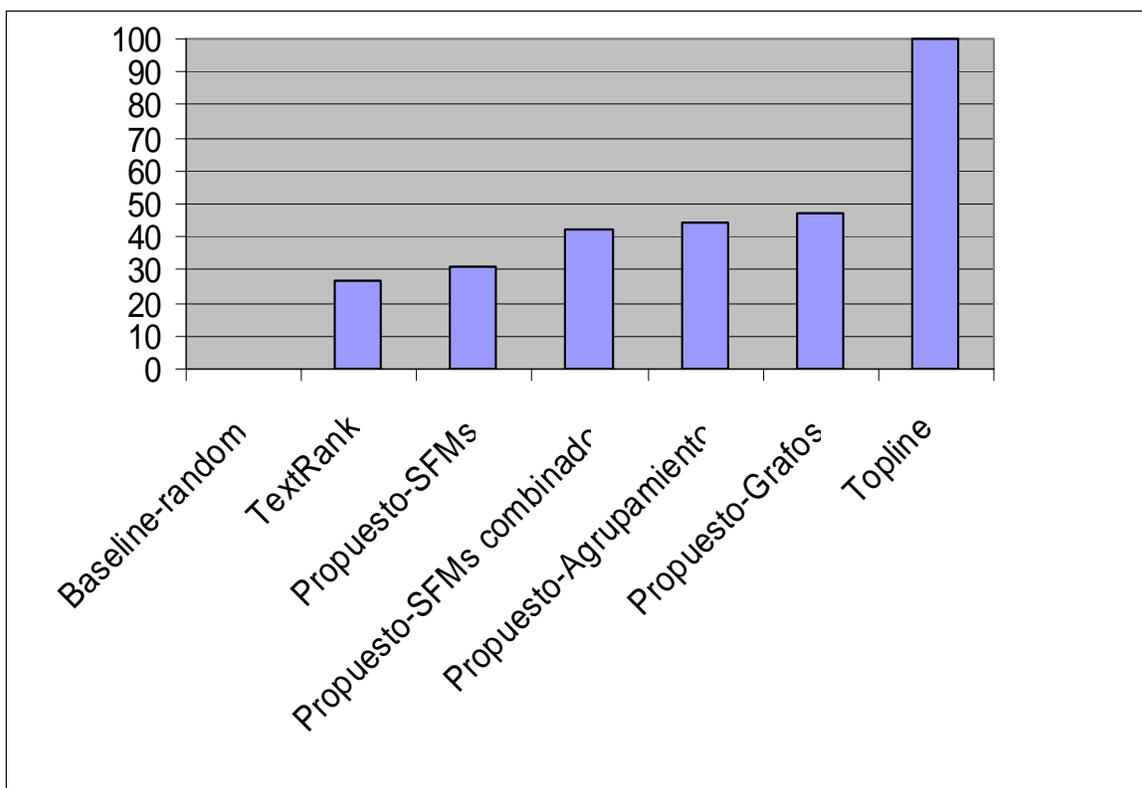
Figure 7.1 – Comparison results for different proposed methods for single document summarization.

## *7.2. Contributions*

In this section we mention the list of contributions, as follows:

– Identification of general steps for an automatic extractive text summarization method.

– Proposal of new methods for generating text summarizes based on the discovery of multiword descriptions.

– Development of the methods for single document summarization.

– Development of the methods for multi document summarization

– New approaches to deal with the task of generation of summaries in a language independent way.

– New approaches to deal with the task of generation of summaries in a dominion independent way.

– **Methods for automatic generation of text summaries which are superior to the state-of-the-art methods.**

– New methods for automatic text summarization using graph, genetic, and clustering algorithms.

– Proposal of new representation for displaying the results of extractive text summarization.

– One of the possible contributions is the application of multiword descriptions for other task of NLP (as future work).

## 7.3. Derived Publications

YULIA LEDENEVA, ALEXANDER GELBUKH, and RENÉ A. GARCÍA HERNÁNDEZ. Terms Derived from Frequent Sequences for Extractive Text Summarization. In: A. Gelbukh (Ed.): CICLing-2008, LNAI 4919, pp. 593-604, February 2008, Springer-Verlag Berlin Heidelberg 2008.

YULIA LEDENEVA, ALEXANDER GELBUKH, and RENÉ A. GARCÍA HERNÁNDEZ. Keeping Maximal Frequent Sequences Facilitates Extractive Summarization. In: Sidorov G., Torres S., etc. (Eds.): CORE-2008, Research in Computing Science, pp. 163-174, May 2008.

YULIA LEDENEVA. Effect of Preprocessing on Extractive Summarization with Maximal Frequent Sequences. In: Gelbukh A., Morales E. (Eds.): MICAI-2008, LNAI, November 2008 (aceptado).

RENÉ A. GARCÍA HERNÁNDEZ, YULIA LEDENEVA, ALEXANDER GELBUKH, et al. Text Summarization by Sentence Extraction using Unsupervised Learning. In: Gelbukh A., Morales E. (Eds.): MICAI-2008, LNAI, November 2008 (aceptado).

**Other Publications during my Ph.D. in CIC-IPN:**

YULIA LEDENEVA, CARLOS A. REYES GARCÍA. Automatic Estimation of Fusion Method Parameters to Reduce Rule Base of Fuzzy Control Complex Systems. A. Gelbukh and C.A. Reyes-Garcia (Eds.): MICAI-2006, Springer-Verlag Berlin Heidelberg 2006, LNAI 4293, pp.146-155, 2006.

YULIA LEDENEVA, CARLOS A. REYES GARCÍA. Automatic Estimation of Sensory Fusion Parameters. Memorias de Séptimo Encuentro de Investigación. INAOE, 8-9 de November, 2006.

YULIA LEDENEVA, CARLOS A. REYES GARCÍA. Automatic Estimation of Parameters for the Hierarchical Reduction of Rules of Complex Fuzzy Controllers. Proceedings of ICINCO-2007, France, May 2007.

YULIA LEDENEVA, ALEXANDER GELBUKH, CARLOS A. REYES GARCÍA, and RENÉ A. GARCÍA HERNÁNDEZ. Genetic Optimization of the Parameters of Fuzzy Control Complex Systems. Torres S. et al (Eds.): CORE-2007, Research in Computing Science ISSN: 1870-4069, pp. 37-48, May 2007.

YULIA LEDENEVA, ALEXANDER GELBUKH, RENÉ A. GARCÍA HERNÁNDEZ, JOSÉ FCO. MARTÍNEZ-TRINIDAD, ARIEL CARRASCO-OCHOA. A New Disagreement Measure for Characterization of Classification Problems. A. Gelbukh (Eds.): CIC-2007, IEEE, November 2007.

**Book Chapters:**

YULIA LEDENEVA, RENÉ A. GARCÍA HERNÁNDEZ, and ALEXANDER GELBUKH. Automatic Estimation of Parameters of Complex Fuzzy Systems. *Robotics, Automation and Control*, I-Tech Education and Publishing (Lazinica A.), Austria, pp. 3-31, (accepted, published in August 2008).

**Invited Talks:**

Como he logrado mi investigación en generación de resúmenes. 8-10 October 2008, Tecnological Institute of Toluca, Mexico.

Generación de resúmenes utilizando descripciones multipalabra. 3 October 2008, INAOE, Mexico.

**Derived Projects:**

Generación de resúmenes de textos por computadora con la detección automática de expresiones multipalabra; proyecto SIP número 20082936, Enero – Diciembre 2008. Director de proyecto Dr. Alexander Gelbukh.

## 7.4. Future Work

We indicate some possible ways and ideas to extend this work:

✓ Although the proposed methods were tested for corpus DUC-2002, the methods can be tested with other corpus to affirm the preliminary conclusions.

✓ Extend single summarization methods.

✓ Test more term weighting, sentence weighting, and sentence selection schemes for multi summarization.

✓ Extract multiword descriptions in order to apply them in other natural language processing tasks.

# *References*

[Agr95] Agrawal R., Srikant R. "Mining Sequential Patterns", In Proc. of the International Conference on Data Engineering, 1995.

[Ace07] Aceves-Peréz R., Montes y Gómez M., Villaseñor Pineda L. Enhancing Cross-Language Question Answering by Combining Multiple Question Translations. Lecture Notes in Computer Science, Springer-Verlag, vol. 4394, pp. 485–493, 2007.

[Bae99] Baeza-Yates R. Modern Informatio Retrieval. Addison Wesley Longman Publishing Co. Inc., 1999.

[Bar99] Barzilay R., Elhadad M. 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, 1999.

[Bar03] Barzilay R. Information Fusion for Multi Document Summarization. Ph.D. thesis. Columbia University, 2003.

[Bar05] Barzilay R., McKeown K. Sentence Fusion for Multi Document News Summarization. Computational Linguistics, Vol. 31, Issue 3, pp. 297–328, ISSN: 0891-2017, 2005.

[Bol01] Bolshakov I., Gelbukh. A. A Very Large Database of Collocations and Semantic Links. Proc. NLDB'2000: 5th International Conference on Applications of Natural Language to Information Systems, France, In: Mokrane Bouzeghoub et al. (eds.) Natural Language Processing and Information Systems. Lecture Notes in Computer Science, ISSN 0302-9743, Springer-Verlag, vol. 1959, pp. 103–114, 2001.

[Bol04a] Bolshakov I., Gelbukh A. Computational Linguistics: Models, Resources, Applications. IPN-UNAM-FCE, ISBN 970-36-0147-2, 2004.

[Bol04b] Bolshakov I. Getting One's First Million... Collocations. Lecture Notes in Computer Science, Springer-Verlag, ISSN 0302-9743, vol. 2945, pp. 226–239, 2004.

[Bol05] Bolshakov I., Galicia-Haro S., Gelbukh A. Detection and Correction of Malapropisms in Spanish by means of Internet Search. 8th International Conference Text, Speech and Dialogue (TSD-2005), Czech Rep. Lecture Notes in Artificial Intelligence (indexed in SCIE), ISSN 0302-9743, ISBN 3-540-28789-2, Springer-Verlag, vol. 3658, pp. 115–122, 2005.

[Bol08] Bolshakov I. Various Criteria of Collocation Cohesion in Internet: Comparison of Resolving Power. Computational Linguistics and Intelligent Text Processing (CICLing-2008, Israel). Lecture Notes in Computer Science, Springer-Verlag, vol. 4919, pp. 64–72, 2008.

[Bri98] Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, vol. 30, pp. 1–7, 1998.

[Bru01] Brunn M., Chali Y., Pinchak C. Text Summarization Using Lexical Chains. Proc. of Document Understanding Conference 2001, http://duc.nist.gov/pubs.html#2001.

[Car98] Carbonell J., Goldstein J. 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, 1998.

[Car03] Carlson L., Marcu D., Okurowski M. E. 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. Kluwer Academic Publishers, 2003.

[Car04] Carberry S., Elzer S., Green N., McCoy K., Chester D. Extending Document Summarization to Information Graphics. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, pp. 3–9.

[Car06] Carberry S., Elzer S., Demir S. Information Graphics: An Untapped Recourse for Digital Libraries. SIGIR'06, ACM 1-59593-369-7/06/0008, 2006.

[Chu04] Chuang T. W., Yang J. Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. Proc. of the ACL-04 Workshop, España, 2004.

[Cha99] Chambers Lance D. Practical Handbook of Genetic Algorithm Complex Coding System, CRC Press, 1999.

[Cha04] Chali Y., Kolla M. Summarization techniques at DUC 2004. Proceedings of the 2004 Document Understanding Workshop Boston, USA, May 6–7, 2004.

[CICLing] CICLing. Conference on Intelligent Text Processing and Computational Linguistics (2000-2009): www.CICLing.org.

[Cri05] Cristea D., et al. Summarization through Discourse Structure. CICLing 2005, LNCS, vol. 3406, Springer-Verlag, pp. 621–632, 2005.

[Cor04] Corston-Oliver S., Ringger E., Gamon M., Campbell R. Task-focused Summarization of emails. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, pp. 43–50.

[Dan03] Daniel N., Radev D., and Allison T. Sub-event based Multi-document Summarization. Proc. of the HLT-NAACL Workshop on Text Summarization, pp. 9–16, 2003.

[Dau04] Daume H., Marcu D. Generic Sentence Fusion and Ill-defined Summarization Task. Proc. of ACL Workshop on Summarization, pp.96–103, 2004.

[Dav07] D'Avanzo E., Elia A., Kuflik T., Vietri S. LAKE System at DUC-2007. Proc. of Document Understanding Conference 2007. http://duc.nist.gov/pubs.html#2007.

[Den06] Denicia-Carral C., Montes-y-Gómez M., Villaseñor-Pineda L., García-Hernández. R. Text Mining Approach for Definition Question Answering, 5th International Conference on Natural Language Processing (FinTal), LNCS, Springer-Verlag 2006.

[Dia06] Dia Q., Shan J. A New Web Page Summarization Method. SIGIR'06, ACM 1-59593-369-7/06/0008, 2006.

[Dor04] Doran W., et al. Assessing the Impact of Lexical Chain Scoring Methods and Sentence Extraction Schemes on Summarization. CICLing, LNCS, vol. 2945, Springer-Verlag, pp. 622–630, 2004.

[Duc] DUC. Document understanding conference; www-nlpir.nist.gov/projects/duc.

[Edm69] Edmundson H.P. New Methods in Automatic Extraction. In Journal of the ACM 16(2), pp. 264–285, 1969.

[Esh91] Eshelman Larry J. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, In: G. Rawlins (Ed.), Foundations of Genetic Algorithms, pp. 265–283, Morgan Kaufmann, 1991.

[Eva05] Evans D., McKeown K. Identifying Similarities and Differences Across Arabic and English News. Proc. of the International Conference on Intelligence Analysis. McLean, VA, 2005.

[Far04] Farzindar A., Lapalme G. Legal Text Summarization by Exploration of the Thematic Structure and Argumentative Roles. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, pp. 27–33.

[Fil04] Filatova E., Hatzivassiloglou V. Event-based Extractive summarization. Proc. of ACL Workshop on Summarization, pp.104–111, 2004.

[Fil07] Filippova K., Mieskes M., Nastase V., et al. Cascaded Filtering for Topic-Driven Multi-Document Summarization. Proc. of Document Understanding Conference 2007. http://duc.nist.gov/pubs.html#2007.

[Fer07] Ferrández S., Ferrández A. The Negative Effect of Machine Translation on Coross–Lingual Question Answering. Lecture Notes in Computer Science, Springer-Verlag, vol. 4394, pp. 494–505, 2007.

[Fut04] Futrelle R. Handling Figures in Document Summarization. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, pp. 61–65.

[Gal07] Galicia Haro S. N., Gelbukh, A. IPN, ISBN 970-36-0265-7, 2007.

[Gar04] García-Hernández R. A., Martínez-Trinidad J. F., and Carrasco-Ochoa J. A. 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-Verlag 2004.

[Gar06] García-Hernández R. A., Martínez-Trinidad J. F., and Carrasco-Ochoa J. A. A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection, CICLing, LNCS, vol. 2945, pp. 514–523, Springer-Verlag 2006.

[Gel03a] Gelbukh A., Sidorov G., Sang Yong Han. Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization. WSEAS Transactions on Communications, ISSN 1109-2742, Issue 1 Vol. 2, pp. 11–19, 2003.

[Gel03b] Gelbukh A., Bolshakov I. Internet, a true friend of translator. International Journal of Translation, ISSN 0970-9819, Vol. 15, No. 2, pp. 31–50, 2003.

[Gel03c] Gelbukh A., Sidorov G., Automatic selection of defining vocabulary in an explanatory dictionary. Proc. CICLing-2002, 3rd International Conference on Intelligent Text Processing and Computational Linguistics, Mexico. Lecture Notes in Computer Science, ISBN 3-540-43219-1, vol. 2276, Springer-Verlag, pp. 300–303, 2002.

[Gel04a] Gelbukh A., Sidorov G., Sang Yong Han, Hernandez-Rubio E. Automatic Enrichment of Very Large Dictionary of Word Combinations on the Basis of Dependency Formalism. In: Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, Humberto Sossa (Eds.). MICAI-2004, Mexican International Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence, Springer-Verlag, ISSN 0302-9743, vol. 2972, pp. 430–437, 2004.

[Gel04b] Gelbukh A., Bolshakov I. On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms. In: Jesus Favela, Ernestina Menasalvas, Edgar Chávez (Eds.) Advances in Web Intelligence (AWIC-2004, 2nd International Atlantic Web Intelligence Conference, Mexico). Lecture Notes in Artificial Intelligence, Springer-Verlag, ISSN 0302-9743, vol. 3034, pp. 105–114, 2004.

[Gel06] Gelbukh A., Sidorov, G. Procesamiento Automático del Español con enfoque en recursos léxicos grandes. IPN, ISBN 970-36-0264-9, 2006.

[Gel08] CV and web-page of Prof. Ph.D. Alexander Gelbukh: www. Gelbukh.com

[Gol89] Goldberg D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.

[Hac05] HaCohen-Kerner Yaakov, et al. Automatic Extraction and Learning of Keyphrases from Scientific Articles. CICLing, LNCS, vol. 3406, Springer-Verlag, pp. 645–657, 2005.

[Hac04] Hachey B., Grover C. A Rhetorical Status Classifier for Legal Text Summarization. Proc. of the ACL-04 Workshop: Text Summarization Branches Out, pp. 35–42.

[Har79] Hartigan J. A., Wong M. A. A K-Means Clustering Algorithm. Applied Statistics, vol. 28(1), pp. 100–108, 1979.

[Hau04] Haupt Randy L., Haupt Sue Ellen. Practical Genetic Algorithms 2nd Edition (ebook), Wiley, 2004.

[Has07] Hassan, S., Mihalcea, R., Banea, C. Random-Walk Term Weighting for Improved Text Classification. Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, 2007.

[Her06a] Hernández-Reyes E., García-Hernández R. A., Martínez-Trinidad J. F., and Carrasco-Ochoa J. A. Document Clustering based on Maximal Frequent Sequences, 5th International Conference on Natural Language Processing (FinTal), LNCS, Springer-Verlag 2006.

[Her06b] Hernández-Reyes E., García-Hernández R. A., Martínez-Trinidad J. F., and Carrasco-Ochoa J. A. Document Representation Based on Maximal Frequent Sequence Sets. 11th Iberoamerican Congress on Pattern Recognition (CIARP), LNCS, Springer-Verlag 2006.

[Hil02] Hardy Hilda: Cross-Document Summarization by Concept Classification. Proc. of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, pp. 121–128, 2002.

[Hov03] Hovy E. The Oxford handbook of Computational Linguistics, Chapter about Text Summarization, In Mitkov R. (ed.), NY, 2003.

[Ina08] El laboratorio de Tecnologías del Lenguaje de la Coordinación de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica, headed by Ph.D. Manuel Montes y Gómez and Ph.D. Luis Villaseñor Pineda: ccc.inaoep.mx/labtl

[Jai99] Jain A.K., Murty M. N., Flynn P. J. Data clustering: a review. ACM Computing Surveys, vol. 31(3), pp. 264–323 1999.

[Kle99] Kleinberg J. Authoritative sources in a hyperlinked enviroment. Journal of the ACM, vol. 46, num. 5, pp. 604–632, 1999.

[Kol05] Kolla M., Chali Y. Experiments in DUC 2005. Proceedings of the 2005 Document Understanding Workshop, Vancouver, Canada, October 9-10, 2005.

[Kos04] Koster, C.H.A. Transducing Text to Multiword Units, Workshop on Multiword Units MEMURA at the 4th International Conference on Language Resources and Evaluation (LREC-2004), Lisbon, Portugal, 2004.

[Kup95] Kupiec J., Pedersen J.O., and Chen F. A Trainable Document Summarizer. Proc. of the 18th ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 68–73, Seattle, 1995.

[Ledo03] Ledo Mezquita Y., Sidorov G., Gelbukh A. Tool for Computer-Aided Spanish Word Sense Disambiguation. Computational Linguistics and Intelligent Text Processing (CICLing-2003, Mexico). Lecture Notes in Computer Science, ISSN 0302-9743, ISBN 3-540-00532-3, Springer-Verlag, vol. 2588, pp. 277–280, 2003.

[Lin97] Lin C. Y., Hovy E. Identify Topics by Position. Proc. of the 5th Conference on Applied NLP, 1997.

[Lin03a] Lin C. Y. ROUGE: A Package for Automatic Evaluation of Summaries. Proc. of the Human Technology Conference (HLT-NAACL), Canada, 2003.

[Lin03b] Lin C. Y., Hovy E. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. Proc. of the Human Technology Conference, Canada, 2003.

[Lin04a] Lin C. Y. Looking for a Few Good Metrics: Automatic Summarization Evaluation – How Many Samples Are Enough? Proc. of NTCIR Workshop, Japan, 2004.

[Lin04b] Lin C. Y., et al. ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In: Proc. of the 20th International Conference on Computational Linguistics (COLING), Switzerland.

[Lin04c] Lin C. Y., et al. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. Proc. of the 42nd Annual Meeting of the ACL, Spain, 2004.

[Liu06a] Liu D., He Yanxiang, and et al. Multi-Document Summarization Based on BE-Vector Clustering. CICLing 2006, LNCS, vol. 3878, Springer-Verlag, pp. 470–479, 2006.

[Liu06b] Liu D., Wang Y., et al. Multiple Documents Summarization Based on Genetic Algorithm. FSKD, LNAI 4223, pp. 355–364, Springer-Verlag 2006.

[Liu06c] Liu D., et al. Genetic Algorithm Based Multi-Document Summarization. PRICAI, LNAI 4099, Springer-Verlag, pp. 1140–1144, 2006.

[Li07] Li J., Sun L. A Query-Focused Multi-Document Summarizer Based on Lexical Chains. Proc. of Document Understanding Conference 2007. http://duc.nist.gov/pubs.html#2007.

[Luh57] Luhn H.P. A statistical Approach to Mechanical Encoding and Searching of literary information. IBM Journal of Research and Development, pp. 309–317, 1957.

[Luh59] Luhn H.P. The automatic Creation of Literature Abstracts. IBM Journal of Research and Development, pp. 159–165, 1959.

[Luh75] Luhn H.P. A Statical Approach to Mechanical Encoding and Searching of Literary Information. IBM Journal of Research and Development, pp. 309-317, 1975.

[Mad07] Madnani N., Zajic D., Dorr B., etc. Multiple Alternative Sentence Compressions for Automatic Text Summarization. Document Understanding Conference 2007. http://duc.nist.gov/pubs.html#2007

[Man99] Manning C. Foundations of Statistical Natural Language Processing, MIT Press, London, 1999.

[Man07] Manning C. An Introduction to Information Retrieval. Cambridge University Press, 2007.

[Mar00a] Marcu D. The Theory and Practice of Discourse and Summarization. The MIT Press, chapter 9, 2000.

[Mar00b] Marcu D. The rhetorical parsing of unrestricted texts: A surface-based approach. Computational Linguistic, vol.26, num. 3, 2000.

[Mar01] Marcu D. Discourse-based summarization in DUC-2001. Document Understanding Conference 2001. http://duc.nist.gov/pubs.html#2001

[Mck03] McKeown K., Barzilay R., Chen J., Elson D., Klavans J., Nenkova A., Schiffman B., Sigelman S. Columbia's Newsblaster: New Features and Future Directions. Proc. of the Human Language Technology Conference, vol. II, 2003.

[Mel99] Melanie Mitchell. An Introduction to Genetic Algorithms (ebook), MIT Press, 1999.

[Men05] Méndez Cruz C., Medina Urrea A. Extractive Summarization Based on Word Information and Sentence Position. CICLing, LNCS, vol. 3406, Springer-Verlag, pp. 641—644, 2005.

[Mih04] Mihalcea, R., Tarau, P. TextRank: Bringing Order into Texts. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Spain, 2004.

[Mih06] Mihalcea R. Ramdom Walks on Text Structures. CICLing 2006, LNCS, vol. 3878, pp. 249–262, Springer-Verlag 2006.

[Mit97] Mitra M., Singhal Amit, Buckley C. Automatic Text Summarization by Paragraph Extraction. ACL/EACL Workshop on Intelligent Scalable Text Summarization, Spain, pp. 31–36, 1997.

[Mon01] Montes y Gómez M., Gelbukh A., López López A. Mining the news: trends, associations, and deviations. Computación y Sistemas, Revista Iberoamericana de Computación, ISSN 1405-5546, Vol. 5, vol. 1, pp. 14–24, 2001.

[Mon02] Montes y Gómez M., Gelbukh A., López López A. Text Mining at Detail Level Using Conceptual Graphs. In: Uta Priss, Dan Corbett, Galia Angelova (Eds.): Conceptual Structures: Integration and Interfaces, 10th International Conference on Conceptual Structures (ICCS), Bulgaria. LNCS, Springer-Verlag, ISSN 0302-9743, vol. 2393, pp. 122–136, 2002.

[Mor91] Morris J., Hirst G. Lexical cohesion computed by thesaurus relations as an indicator of the structure of text. *Computational Linguistics*, vol. 18, pp. 21–45, 1991.

[Neg02] Negnevitsky M. Artificial Intelligence: A Guide to Intelligent Systems, Addison Wesley, Harlow, England, 2002.

[Nen04] Nenkova A., Passonneau R. Evaluating content selection in summarization: The pyramid method. In: Proc. of NLT/NAACL–2004, 2004.

[Nen05a] Nenkova A., Siddharthan A., McKeown K. Automatically Learning Cognitive Status for Multi-Document Summarization of Newswire. Proc. of HLT/EMNLP-05, 2005.

[Nen05b] Nenkova A., Vanderwende L. The Impact of Frequency on Summarization No. MSR-TR-2005-101). Redmond, Washington: Microsoft Research, 2005.

[Nen06] Nenkova A. Understanding the process of multi-document summarization: content selection, rewriting and evaluation. Ph.D. Thesis, Columbia University, 2006.

[Net04] Neto L., Freitas A., et al. Automatic Text Summarization using a Machine Learning Approach. Proc. of the ACL Workshop, España, 2004.

[Nad06] Nadjet Bouayad-Agha, and et al. A sentence compression module for machine-assisted subtitling. CICLing, LNCS, vol. 3878, Springer-Verlag 2006.

[Ora05] Orasan C. Automatic Annotation of Corpora for Text Summarization: A Comparative Study. CICLing, LNCS, vol. 3406, Springer-Verlag, pp. 658–669, 2005.

[Ott06] Otterbacher J., Radev D., Kareem O. News to Go: Hierarchical Text Summarization for Mobile Devices. SIGIR'06, ACM 1-59593-369-7/06/0008.

[Pas05] Passonneau R., Nenkova A., McKeown K., Sigleman S. Pyramid evaluation at DUC-05. Proc. of Document Understanding Conference, http://duc.nist.gov/pubs.html#2007.

[Por80] Porter M.F. An algorithm for suffix stripping, Program, 14(3), pp. 130–137, 1980.

[Rad03] Radev D., Tam D. Single-document and multi-document summary evaluation via relative utility. Proc. of the ACM International Conference on Information and Knowledge Management (CIKM-03), 2003.

[Rad04] Radev D., Jing Hongyan, Budzikowska Malgorzata: Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation and User Studies. Information Processing and Management, vol. 40, pp. 919–938, 2004.

[Ree07] Reeve L. H., Han H. A Term Frequency Distribution Approach for the DUC-2007 Update Task. Proc. of Document Understanding Conference 2007.
http://duc.nist.gov/pubs.html#2007.

[Sal88] Salton G., Buckley C. Term-Weighting Approaches in Automatic Text Retreival, Information processing and Management, vol. 24, pp. 513–523, 1988.

[Sal89] Salton G. Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer. Addison Wesley Publishing Company, 1989.

[Sar05] Sarkar K., et al. Generating Headline Summary from a Document Set. CICLing, LNCS, vol. 3406, Springer-Verlag, pp. 637–640, 2005.

[Sek02] Seki Y. Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles. Proc. of the Third NTCIR Workshop, 2002.

[Shr04] Shrestha L., McKeown K. Detection of Question-Answer Pairs in Email Conversation. Proc. of Coling 2004, Switzerland, 2004.

[Sid05] Sidorov G. Proyecto CONACYT: Diccionarios dinámicos orientados al dominio como una heramienta para medición y representación de tendencias de desarrollo de disciplinas científicas y tecnologias y inerracciones entre ellos, 2002-2005.

[Sil02] Silber H., McCoy K. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4), pp. 487–496, 2002.

[Son04] Song Y., et al. A Term Weighting Method based on Lexical Chain for Automatic Summarization. CICLing 2004, LNCS, vol. 32945, Springer-Verlag, pp. 631–634, 2004.

[Sor03] Soricut R., Marcu D. Sentence level discourse parsing using sintactic and lexical information. In: HLT-NAACL, 2003.

[Teu04a] Teusel S., Halteren H. van. Agreement in human factoid annotation for summarization evaluation. Proc. of the 4th Conference on Language Resources and Evaluation (LREC), 2004.

[Teu04b] Teusel S., Halteren H. van. Evaluating Information content by factoid analysis: human annotation and stability. EMNLP, pp. 419–426, 2004.

[Una08] Group of NLP of Universidad Nacional Autónoma de México headed by Galicia Haro Sofia N.

[Una05] Una-May O´Reilly, et al. (Eds.) Genetic Programming Theory and Practice II, Springer, 2005.

[Van04] Vandeghinste V., Pan Y. Sentence Compression for Automated Subtitling: A Hybrid Approach. Proc. of ACL Workshop on Summarization, pp. 89–95, 2004.

[Ver04] Verbeek Jakob. Mixture Models for Clustering Dimension Reduction. Thesis, ISBN 90-776-125-4, 2004.

[Ver05] Vergara-Villegas O., García-Hernández R. A., Carrasco-Ochoa J. A., Pinto R., Martínez-Trinidad J. F. Data Preprocessing by Sequential Pattern Mining for LZW, 6th Mexican International Conference on Computer Science (ENC), IEEE Computer Society Press, 2005.

[Ver07] Verma R., Chen P. A Semantic Free-text Summarization System Using Ontology Knowledge. Proc. of Document Understanding Conference 2007.
http://duc.nist.gov/pubs.html#2007.

[Vil03] Villaseñor-Pineda L., Montes-y-Gómez M., Vaufreydaz D., Serignat J-F. Elaboración de un Corpus Balanceado para el Cálculo de Modelos Acústicos usando la Web. Proc. of the International Conference on Computing, 2003.

[Vil04] Villaseñor-Pineda L., Montes-y-Gómez M., Vaufreydaz D., Serignat J-F. Experiments on the Construction of a Phonetically Balanced Corpus from the Web. Lecture Notes in Computer Science, ISSN 0302-9743, Springer-Verlag, vol. 2945, pp. 410–413, 2004.

[Vil06] Villatoro-Tello E., Villaseñor- Pineda L., and Montes-y-Gómez M. Using Word Sequences for Text Summarization. 9th International Conference on Text, Speech and Dialogue (TSD). Lecture Notes in Artificial Intelligence, Springer-Verlag, 2006.

[Voo04] Voorhees Ellen M. Overview of the TREC-2003 question answering task. In: Proc. of the 12th Text Retrieval Conference (TREC-2003), pp. 54–68, 2004.

[Wan04] Wan S., McKeown K. Generating "State-of Affairs" Summaries of Ongoing Email Thread Discussions. Proc. of Coling 2004, Switzerland, 2004.

[Wan05] Wang R., et al. LexTrim: A Lexical Cohesion based Approach to Parse-and-Trim Style Headline Generation. CICLing 2005, LNCS, vol. 3406, Springer-Verlag, pp. 633–636, 2005.

[Wan06] Wan X., Yang J., et al. Incorporating Cross-Document Relationships Between Sentences for Single Document Summarizations. ECDL, LNCS, vol. 4172, Springer-Verlag, pp. 403–414, 2006.

[Wei06] Xu Wei, Li Wenjie, et al. Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. CICLing 2006, LNCS, vol. 3878, Springer-Verlag, pp. 480–489, 2006.

[Yin07] Ying J.-C., Yen S.-J., Lee Y.-S. Language Model Passage Retrieval for Question-Oriented Multi Document Summarization. Proc. of Document Understanding Conference 2007. http://duc.nist.gov/pubs.html#2007.

[Zho05] Zhou Q., Sun L. IS_SUM: A Multi-Document Summarizer based on Document Graphics and Lexical Chains. Proc. of Document Understanding Conference 2005. http://duc.nist.gov/pubs.html#2005.

# *Appendix 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 | |

# *Appendix B. Examples of Results*

Detailed (complete) results for the best three results from Table 5.2 (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, 1best+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)

# Appendix C. Examples of Multiword Descriptions

Flights were cancelled

Sunday night

Prensa Latina

Civil defence

Hurricane Gilbert

The Dominican Republic

The south coast

The national weather service said

The Cayman Islands

Cancun and Cozumel

In Mexico City

Quintana Roo state

Over the water

Roamed the streets of Cancun

The Yucatan peninsula

Tropical storm

Low pressure

Caused coastal flooding

San Francisco area

Have to pay

Have earthquake insurance

Insurance companies

Long term

A special session

To deal with

Department of transportation

Gasoline tax increase

Might collapse in an earthquake

The white house

The California earthquake

Bush and his aides

The insurance industry

Exposure to catastrophes

On an inflation adjusted basis

Structural damage

State farm

Personal property

Earthquake insurance

Plenty of experience

Year after year

For the purpose of

Whenever I needed him

The royal marine's music school

Military installations

Private security

Irish republican army

Opening day record for

Restaurant in Moscow

The Soviet Union

The big mac

Moscow McDonalds

Above and beyond the usual guest

Vice president of the United States

Wal mart discount city

Most important retailer of his generation

British Prime Minister John Major

Vote for Major

Major was elected

Was elected to parliament

Leader of the Conservative Party

Associated Press

# *Appendix D. Examples of Generated Summaries*

**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 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 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 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.

**System Summary:**

(was obtained automatically using the proposed method from 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 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 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.

**System Summary:**

(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.