

University of South Wales



2059769

Bound by

 **ABBEY** BOOKBINDING  
& PRINTING

Unit 3 Gabalfa Workshops Excelsior Ind. Est. Cardiff CF14 3AY  
Tel: (029) 2062 3290 Fax: (029) 2062 5420  
Email: [info@abbeybookbinding.co.uk](mailto:info@abbeybookbinding.co.uk)  
Web: [www.abbeybookbinding.co.uk](http://www.abbeybookbinding.co.uk)

# **SEMANTIC BASED CONTENT SEARCH AND CONTENT SUMMARIZATION**

**GEORGIOS MAMAKIS**

**A submission presented in partial fulfilment of the requirements of the  
University of Glamorgan/Prifysgol Morgannwg for the degree of Doctor  
of Philosophy**

**October 2012**

---

## Table of Contents

Table of Figures .....	v
List of Tables .....	vi
Abstract .....	vii
Acknowledgements .....	viii
Certificate of Research .....	ix
List of Acronyms .....	x
1. Introduction .....	1
1.1 Document Classification .....	1
1.2 Document Summarization .....	3
1.3 Motivation .....	5
1.4 Research Aims and Objectives .....	5
1.5 Software Utilised .....	6
1.6 Overview of the Summarization Processes .....	7
1.6.1 TCASGL .....	7
1.6.2 GUTS .....	8
1.7 Thesis Outline .....	10
2. Related Work on Statistical Document Classification .....	12
2.1 Introduction .....	12
2.2 Naive Bayes Classifiers .....	12
2.2.1 Naive Bayes Classifier Efficiency .....	14
2.3 Language Models .....	15
2.4 Synopsis .....	17
3. Related Work on Document Summarization .....	18
3.1 Document Summarization in General .....	18
3.2 Early Generic Single-Document Summarization .....	18
3.2.1 Statistical Approaches .....	19
3.2.2 Machine Learning Approaches .....	23
3.2.3 Natural Language Processing Approaches .....	26
3.3 Modern Approaches in Generic Single-Document Summarization .....	29
3.3.1 Machine Learning Approaches .....	29
3.3.2 Natural Language Processing Approaches .....	33

---

3.4	Query-Based and Multidocument Summarization .....	39
3.5	Synopsis.....	42
4.	Linguistic Analysis for TCASGL and GUTS.....	43
4.1	Nouns, Adjectives and Their Importance .....	43
4.2	Syntactical Analysis – Part of Speech Tagging.....	43
4.3	Grammatical Analysis- Stemming.....	44
4.4	Grammar or Syntax? .....	44
4.5	Synopsis.....	47
5.	Text Classification Assisted Summarization for Greek Language - TCASGL ....	48
5.1	Introduction .....	48
5.2	TCASGL approach.....	50
5.3	TCASGL Methodology.....	53
5.3.1	Training Step .....	54
5.3.2	Classification Step .....	55
5.3.3	Summarization Step .....	56
5.4	Synopsis.....	56
6.	Generic Unsupervised Text Summarization - GUTS .....	58
6.1	Introduction .....	58
6.2	GUTS Approach.....	60
6.2.1	Semantic lexicon creation .....	60
6.2.2	Conceptual Flow Topic Creation and Summarization .....	61
6.2.3	Intra-document Summarization.....	63
6.3	Methodology .....	63
6.4	Synopsis.....	69
7.	Evaluation – Approaches and Results.....	70
7.1	Document Summarization Evaluation .....	70
7.2	Intrinsic Evaluation .....	70
7.2.1	Precision, Recall, F-measure.....	71
7.2.2	ROUGE-N .....	72
7.2.3	Pyramid.....	72
7.2.4	Basic Elements .....	73
7.3	Extrinsic Evaluation .....	73
7.3.1	Usefulness and Responsiveness.....	74
7.3.2	Relevance-Prediction .....	74

---



---

7.3.3	Complex Approaches .....	75
7.4	Document Summarization Conferences and Workshops .....	75
7.5	Evaluation Tools for Current Research .....	78
7.5.1	TCASGL Evaluation approach .....	78
7.5.2	GUTS Evaluation Approach .....	82
7.6	TCASGL Evaluation Results .....	82
7.6.1	Classification Module .....	82
7.6.2	TCASGL Summarization Module Evaluation.....	85
7.6.3	TCASGL Efficiency.....	87
7.7	GUTS.....	88
7.7.1	GUTS Example .....	88
7.7.2	Automatic Evaluation of Results .....	89
7.7.3	Manual Evaluation Results .....	90
7.7.4	GUTS Efficiency.....	91
7.8	Human Summarization Approach .....	91
7.9	Synopsis.....	92
8.	Conclusions and Future Work.....	94
8.1	Introduction .....	94
8.2	Overview of Research .....	95
8.2.1	Stemming.....	95
8.2.2	Document Classification .....	95
8.2.3	Machine Learning Summarization.....	96
8.2.4	Natural Language Processing Summarization.....	96
8.3	Overview of TCASGL Methodology .....	96
8.4	Overview of GUTS Methodology .....	98
8.5	Contribution to Knowledge.....	99
8.6	Limitations.....	100
8.7	Future Work .....	101
8.8	Final Remarks .....	102
	<u>References</u> .....	103
	<u>APPENDIX A – Noun and Verb Endings and Stop-Lists</u> .....	113
	Noun and Verb Endings.....	113
	Stop-List .....	113
	<u>APPENDIX B - Data Cd</u> .....	115

---

---

<u>APPENDIX C – Published Papers</u> .....	116
--	-----

---

## Table of Figures

Figure 1. TCASGL Training Phase .....	8
Figure 2. TCASGL Summarization Phase .....	8
Figure 3. GUTS Summarization Phase .....	10
Figure 4. Stemming and Noun/Adjective Identification .....	46
Figure 5. TCASGL Overview.....	54
Figure 6. Training Module Algorithm.....	55
Figure 7. Classification Step.....	56
Figure 8. Summarization Module Algorithm .....	56
Figure 9. GUTS Overview.....	64
Figure 10. Abstract Semantic Relation Matrix Creation.....	65
Figure 11. Conceptual Flow Topic Cluster Identification .....	67
Figure 12. Topic Cluster Summarization Module.....	68
Figure 13. Intra-Document Topic Cluster Identification module .....	68
Figure 14. Term Density Summarization module.....	69
Figure 15. Original Document and TCASGL Summary – code a14.....	86
Figure 16. GUTS Summary – code a14 .....	88
Figure 17. Jaccard Similarity – a14 document.....	88

---

## List of Tables

Table 1. Greek Verbs .....	45
Table 2. Weight of word “πολιτικ” in every category .....	52
Table 3. Nouns per category .....	79
Table 4. Overall Classification Results .....	83
Table 5. Classification Results per Category .....	83
Table 6. Classification with ambiguous input data .....	85
Table 7 TCASGL without Positional Characteristics .....	87
Table 8. TCASGL without Positional Characteristics .....	87
Table 9. GUTS without positional characteristics ROUGE-1 score .....	89
Table 10. GUTS with positional characteristics ROUGE-1 score.....	90
Table 11. Human Evaluation Results Normalized.....	90

---

## Abstract

Document summarization has been an intriguing task of Computational linguistics. A number of definitions have been proposed in References, all of which consider document summarization as a problem of text compression. One of the most complete definitions by Sparck-Jones states that "...a summary is a reductive transformation of source text to summary text through content condensation by selection and/or generalisation on what is important in the source...". The importance of document summarization does not lie only in presenting information in a shortened form, but also in selecting the most appropriate content to present. Moreover, a main feature in summarization is the number of sources from which a summary may be produced; thus, single-document and multi-document have been proposed, denoting the number of sources from which the summary will be produced. In addition, another categorization that may be extracted from this definition refers to the importance of the source, and what the potential user thinks is important. This leads to the definition of generic and query-based or task focused summarization, where generic implies that the summarizer should extract information according to the main topics discussed in the document, while query-based summarization focuses on extracting information according to simple or more complex questions on the document. Moreover, importance of content can be extracted through knowledge-rich (supervised and semi-supervised summarization) and knowledge lean approaches (unsupervised or shallow summarization). The last categorization refers to the type generation of the summary, the two main categories being: extractive summarization, where sentences are maintained in the summarization process unaltered; and abstraction, where the sentences are either semantically altered or compressed.

The research depicted in this thesis, presents novel document summarization approaches based on the theories of Machine Learning (ML) and Natural Language Processing (NLP) for generic single-document extractive summarization. The motivation to target on Greek language came from the lack of a Greek summarization system. Most notably, only one system for Greek Summarization system exists in the literature (GreekSum). The research undertaken resulted in: the development of a stemming algorithm used for noun and adjective identification, based on grammatical analysis on Greek language; the development of a novel statistical classification scheme, initially aimed to document summarization, that is proven to outperform other statistical summarizers as Naïve Bayes Classifier (NBC) and Language Models (LM); the development of a supervised statistical summarization algorithm based on document classification techniques (Text Classification Assisted Summarization for Greek Language-TCASGL); and the development of a knowledge-lean summarization algorithm (Generic Unsupervised Text Summarization - GUTS), using shallow semantic document analysis and statistics. The results demonstrate that the classification algorithm significantly outperforms widely available statistical algorithms, while the ML approach yielded comparable results to other supervised systems. In addition to that, GUTS was shown to perform equally well with knowledge rich approaches.

---

## **Acknowledgements**

Initially, I would like to thank my supervisory team at the University of Glamorgan and the Technological Educational Institute of Crete for offering me fruitful guidance and assistance in tackling a series of problems during my research. Especially, I would like to thank Athanasios Malamos, Andrew Ware and Georgios Papadourakis.

Moreover, I would like to thank my family and friends who supported and tolerated with me in the hard times of my research, and shared my vision and thoughts. Especially, I would like to thank my parents for their support and understanding throughout my research.

In addition to that, I would like to thank the laboratory members of Multimedia Content Laboratory of TEI Crete for their invaluable assistance in key points of my research, and the philologist Ioanna Karelli for her assistance in the evaluation of my research.

Special thanks will have to go to Dimitrios Karayannakis of the Technological Institute of Crete for his supportive spirit and extensive help in and directives on special issues that arose during research.



## Certificate of Research

*This is to certify that, except where specific reference is made, the work described in this thesis is the result of the candidate's research. Neither this thesis, nor any part of it, has been presented, or is currently submitted, in candidature for any degree at any other University.*

Signed ..... G. MAMAKIS .....

Georgios Mamakis - Candidate

Date ..... 30/12/2012 .....

Signed ..... J Andrew Ware .....

J Andrew Ware - Director of Studies

Date ..... 30 DECEMBER 2012 .....

---

## List of Acronyms

**AI:** Artificial Intelligence  
**API:** Application Programming Interface  
**BE:** Basic Elements  
**CEM:** Classification Expectation Maximization  
**CML:** Classification Maximum Likelihood  
**CRF:** Conditional Random Fields  
**DMS:** Discourse Macro Structure  
**DUC:** Document Understanding Conference  
**FFNN:** Feed Forward Neural Network  
**GA:** Genetic Algorithm  
**GUTS:** Generic Unsupervised Text Summarization  
**HMM:** Hidden Markov Model  
**IR:** Information Retrieval  
**kNN:** k-Nearest Neighbour  
**LM:** Language Models  
**LSA:** Latent Semantic Analysis  
**MAP:** Maximum A Posteriori Decision Rule  
**ML:** Machine Learning  
**MMR:** Maximal Marginal Relevance  
**MUC:** Message Understanding Conference  
**NBC:** Naive Bayes Classifier  
**NLP:** Natural Language Processing  
**NN:** Neural Network  
**OPP:** Optimum Position Policy  
**PLSI:** Probabilistic Latent Semantic Indexing  
**PNN:** Probabilistic Neural Network  
**POS:** Part of Speech  
**ROUGE:** Recall Oriented Understudy for Gisting Evaluation  
**RST:** Rhetorical Structure Theory  
**SCU:** Summary Content Unit  
**SVD:** Singular Value Decomposition  
**SVM:** Support Vector Machine  
**SVO:** Subject Verb Object  
**TCASGL:** Text Classification Assisted Summarization for the Greek Language  
**tf.idf:** Term Frequency and Inverse Document Frequency  
**tl.tf:** Term Length and Term Frequency  
**ToC:** Table of Contents



## 1. Introduction

*Chapter 1 provides general information on the scope of research and an insight on document classification and summarization. In addition, a formulation of the problems of summarization and classification is described, as well as the key issues that need to be tackled in both problems. The aims and objectives are also provided and justified and reference to the software used is presented. The chapter concludes with an outline of the systems proposed as well as an outline of the remainder of the thesis.*

---

### 1.1 Document Classification

Text classification or text categorization is the task of assigning a random document to a class (single-label classification) or a number of classes (multi-label classification) retrieved from a pre-defined set of possible categories. A special case of single-label classification is binary classification, where the systems choose between two possible classes. Text classification may be applied to numerous areas, such as email spam filtering (binary classification) and context identification (multi-label classification) among others.

Numerous approaches have been proposed to achieve such a task including among others statistics, vector space models, artificial intelligence, decision trees and rule-based methods. One of the simplest approaches is statistical classifiers. The initial assumption of statistical classifiers is the exploitation of observed features that are present in a document, such as word or character occurrence. Statistical classifiers, despite their simplicity and naive assumptions have proven to achieve great results, outperforming in many cases more complex algorithms on a speed-efficiency trade-off (Konstantis & Pintelas, 2004). Another positive characteristic is their speed of execution compared with other more complex approaches (for example, Support

Vector Machines - SVMs), which allows them to be used for real-time applications (Tsoumakas, Katakis, & Vlahavas, 2006). Work on statistical methods for text classification such as Naive Bayes Classifiers (NBC) has been undertaken with significant results regarding its simplicity and efficiency (Rennie, Shih, & Karger, 2003), (Rish, 2001), while statistical learning models have often initiated interest either as an autonomous statistical approach (Srikanth & Srihari, 2002), (Croft, 2003), (Ponte & Croft, 1998) alternate to NBC or in conjunction to NBC to achieve better results (Peng, Schuurmans, & Wang, 2004). Apart from statistical methods another common approach is the definition of vector space models, utilizing algorithms such as k-Nearest Neighbour (kNN) (Manning, Raghavan, & Schütze, 2008). These algorithms try to identify the similarities of random documents based on a 2D representation of training data either by approximating similarity based on proximity information of a random document through its pre-classified neighbours (kNN), or by visualizing a 2D space split by lines, planes or hyper-planes, denoting the classes a random document may belong to, and trying to fit the document into the most appropriate class based on word similarity through vector distance (Rocchio algorithm) (Rocchio, 1971). Other approaches include artificial intelligence classifiers (Neural Networks) as in (Ripley, 1994). Latest approaches in the area of statistical document classification introduce the use of keyphrases as in (Karanikolas & Skourlas, 2010). A keyphrase according to the authors is a set of words commonly found in pre-classified documents, provided that they are found in more than one classes. The authors try to identify the existence of keyphrases in random documents, comparing them either on a document level (to identify the similarity between the random document and any pre-classified document), or on a class level (to identify the similarity between the random document and a class).

A common approach shared by all these classification methods includes a training process, where the corresponding systems use a pre-classified training example set of words or documents, and a test dataset, where the efficiency of the systems is estimated, after the training phase. Online corpora to assist in that direction exist, such

as the TREC (TREC Corpus) and Reuters (Reuters Corpus) corpora for the English language.

## **1.2 Document Summarization**

Document summarization has been a very intriguing area of content management. An excellent attempt to overview document summarization methods has been made by (Sparck Jones, 2007). In this attempt the author focused generally on automatic document summarization techniques, both single and multi-document, generic and query based, omitting a series of algorithms based on shallow semantic analysis on single document summarization.

A complete definition on document summarization has been provided by (Sparck Jones, 2007), where the author stated that “...a summary is a reductive transformation of source text to summary text through content condensation by selection and/or generalisation on what is important in the source...”. As it may become obvious from the definition the importance of document summarization does not lie only in presenting information on a shortened form, but also in selecting the most appropriate content to present. Moreover, a main feature in summarization is the number of sources from which a summary may be produced; thus, single-document and multi-document have been proposed, denoting the number of sources from which the summary will be produced. In addition to that, another categorization that may be extracted from this definition refers to the importance of the source, and what the potential user thinks is important. This leads to the definition of generic and query-based or task focused summarization, where generic implies that the summarizer should extract information according to the main topics discussed in the document, while query-based summarization focuses on extracting information according to simple or more complex questions on the document. Moreover, importance of content can be extracted through knowledge-rich (supervised and semi-supervised summarization) and knowledge lean approaches (unsupervised or shallow summarization). The last categorization refers to the type generation of the summary, the two main categories being extractive summarization, where sentences are

maintained in the summarization process unaltered and abstraction, where the sentences are either semantically altered or compressed.

Automatic summarization can be analyzed into three distinctive tasks according to (Lin & Hovy, 2000): a) topic identification – identifying the topics that are included in a random document, b) topic interpretation - understanding and fusing the documents into more generic ideas, c) summary generation - extracting the summary from the original sentences. A more generic approach on the steps required are the identification of topics, the identification of relevant sentences, words or topic segments that form up the topics and the identification of sentence significance for inclusion in the final summary. These steps have been followed in almost any approach presented in this thesis. Without loss of generalization, it can be stated that regardless of the summarization technique utilized, a system should consider all three distinctive summarization tasks. First of all, it is important to evaluate the document features that signify a potential document topic (topic identification). Topic identification may result from high frequency words or sentence position as in shallow statistical approaches or document training on word importance in more complicated Machine Learning (ML) approaches or even from term co-occurrence or semantics in cases of deep Natural Language Processing (NLP) techniques. Moreover, the identification of such text segments enables the clustering of related information either in the form of generalization (Lin & Hovy, 2000) or through extensive discourse structure analysis on the document which results in rhetoric or discourse trees, latent semantic mapping of topic hierarchies or lexical chaining techniques. The last assumption is that, evaluating sentence significance enables the isolation of content rich text segments and their extraction to form the final summary. Therefore, regardless of the technique used, it may be observed that the metrics taken into consideration while analyzing a document, target almost exclusively in the assignment of an appropriate significance score to each of the text segments that form a document, that will enable the conceptual extraction of significant information.

Initial approaches, included shallow analysis on empirical text features such as term frequencies (Luhn, 1958), sentence positions (Baxendale, 1958), key phrases or cue

phrases (Edmundson, 1969). More complicated approaches have also been proposed that exploit machine learning techniques, natural language analysis, as well as artificial intelligence.

### **1.3 Motivation**

The motivation behind this research stemmed from the lack of existing systems that perform document summarization in Greek language. More specifically, only one system exists that performs document summarization (Pachantouris, 2005). This system comes from a direct translation from Swedish language to the Greek language, therefore missing out the fact that there are particularities in Greek language that may be exploited in performing better summarization tasks. In addition to that, it focuses on single document summarization (summarization from single document sources) rather than multidocument summarization, in order to form the basis upon which Greek multi-document summarization techniques can be extended in the future.

### **1.4 Research Aims and Objectives**

The following aims were identified prior to research.

- To research and propose new document summarization techniques, algorithms and systems based on the current state-of-the-art methodologies, as well as newly identified features.
- To identify whether the exploitation of language-specific linguistic characteristics can facilitate in performing better document summarization tasks.

This research analyses the generic single-document extractive summarization for the Greek language. The selection of the Greek language was not random, since only one system has been developed with the aim of facilitating Greek language summarization, which is a translation of a trainable Swedish language summarizer. Research targets on three general approaches, linguistics, ML and NLP.

Prior to identifying the research objectives the following remarks were made while researching the available literature:

- In ML approaches, document summarization is cut off from document classification, even though classification has been considered as a sentence selection criterion. However, as it has been stated by (Barzilay & Lee, 2004), in document summarization it is important to identify the document class, as each distinct category of topics may have its own terminology.
- NLP approaches initially focused on quantitative or positional characteristics of the document in order to identify document topics and extract document sentences, while latter approaches use complex mathematical approaches to evaluate qualitative characteristics as the diaspora and importance of topics in the document. However, none of these methodologies considered language-specific characteristics. In addition to that, a novel feature was identified that had not been exploited, based on the document topic distribution.

These remarks led to the the formulation of the following research objectives that facilitate the research aims:

- Develop a ML system and methodology that will assist in identifying whether document classification for terminology extraction can assist document classification
- Develop a methodology and system that will combine already explored and novel linguistic and document features, with regards to efficiency.

While developing the ML system, a novel statistical document classification methodology and system was introduced that outperformed other statistical algorithms. In addition to that, an extension was made to the Greek stemming methodology proposed by (Kalamboukis, 1995).

### **1.5 Software Utilised**

The software used for the development of the systems of GUTS and TCASGL, were Oracle Java 1.6 EE (Java), various versions of Oracle Netbeans platform (Netbeans) – the latest being version 7.0. The training data set for the Classification Module of TCASGL was stored in MySQL 5.1.33 (MySQL Community Edition) database.

For the evaluation of the classification module of TCASGL, the Naïve Bayes Classifier Implementation included in Mallet software (McCallum) has been used, while comparisons against Language Models were accomplished through the implementation provided by Lingpipe (Alias-I). Comparisons for the summarization module were carried out, using the GreekSum online adaptation of SweSum available at <http://www.nada.kth.se/iplab/hlt/greeksum/index.htm>, as well as the LSA approach (Gong & Liu, 2001) using an SVD Java library available by JAMA (Java Matrix package) (Hicklin, Moler, Webb, Boisvert, Miller, & Remington).

The training and test data sets were manually gathered from a number of online newspaper sites. These are: [www.contra.gr](http://www.contra.gr), [www.imerisia.gr](http://www.imerisia.gr), [www.naftemporiki.gr](http://www.naftemporiki.gr), [www.tovima.gr](http://www.tovima.gr), [www.in.gr](http://www.in.gr), [www.sport24.gr](http://www.sport24.gr). The data sets were gathered over a time period of 5 years, in order to ensure a great variety of authoring styles. This eliminates the possibility of adapting the methodology to a specific style of writing or to a specific trend topic.

## **1.6 Overview of the Summarization Processes**

The outcome of this research is two algorithms and the according systems directly fulfilling the research objectives. The first system is an ML-based document classification and summarization system (TCASGL) that exploits a novel statistical classification methodology that was proven to outperform other statistical classification algorithms. In addition to that, a novel NLP methodology and system (GUTS) that introduces a sentence clustering criterion (conceptual flow) and considers a number of already researched sentence selection criteria (sentences position, term frequency etc). The following sections provide an outline on the systems and methodologies developed during research.

### **1.6.1 TCASGL**

Text Classification-Assisted Summarization for Greek Language (TCASGL) consists of 4 modules: a stemmer, a training module, a classification module and a summarization module. The stemmer is used whenever a new document is provided to the system, regardless if it is used for training, classification or summarization, and its

aim is to filter out surplus text information, gathering only nouns and adjective. The training module takes the stemmed document words and assigns a score on each of the supplied document word per class. The classification module takes the stemmed document words and decides on the class this document may belong to, from the ones it can identify. The summarization module takes the stemmed document and document class and statistically extracts the most significant sentences using the class dictionary. The training and summarization phase can be depicted in Figures 1 and 2 respectively. More information on the TCASGL system is provided in Chapter 5.

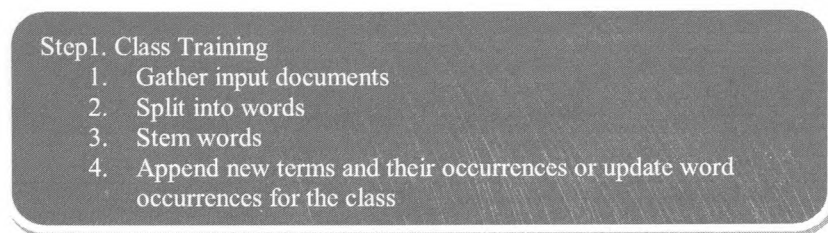


Figure 1. TCASGL Training Phase

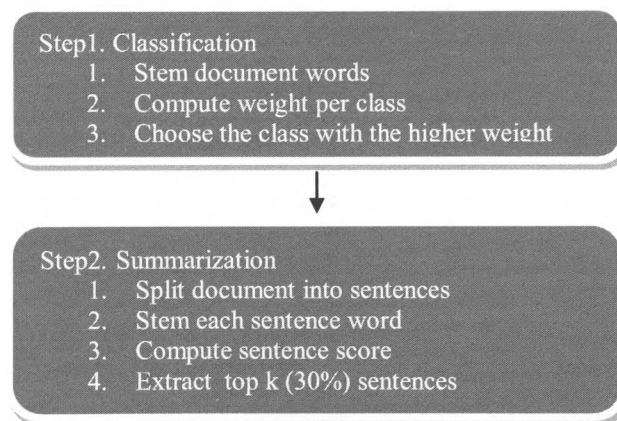


Figure 2. TCASGL Summarization Phase

### 1.6.2 GUTS

Generic Unsupervised Text Summarization (GUTS) system consists of six modules: a stemmer, an abstract semantic lexicon creator, a conceptual flow cluster evaluator, an intra-document cluster evaluator, a topic cluster summarizer and a statistical summarizer. The stemmer used is common between TCASGL and GUTS. The abstract semantic lexicon creator performs an analysis on document co-occurrence and constructs a semantic matrix of the document. The conceptual flow cluster evaluator



identifies the flow of concepts on consecutive sentences and topic shifts and constructs thematic clusters which are not verified on topic similarity. The intra-document cluster evaluator considers cross-document topic clustering, to identify topic repetition and constructs the final topic hierarchy of the document. This step is performed after the initial summarization of the conceptual flow topic clusters. The topic cluster summarizer extracts the most salient sentences from each a topic cluster hierarchy, provided either from the conceptual flow or from the intra-document cluster evaluators. The statistical summarizer ensures that if the topic cluster evaluators fail to identify any topic cluster, or the number of topic clusters is greater than the summary length, the desired length of summary will be reached. The schematic representation of the algorithm is depicted in Figure 3. More information on GUTS system is provided in Chapter 6.

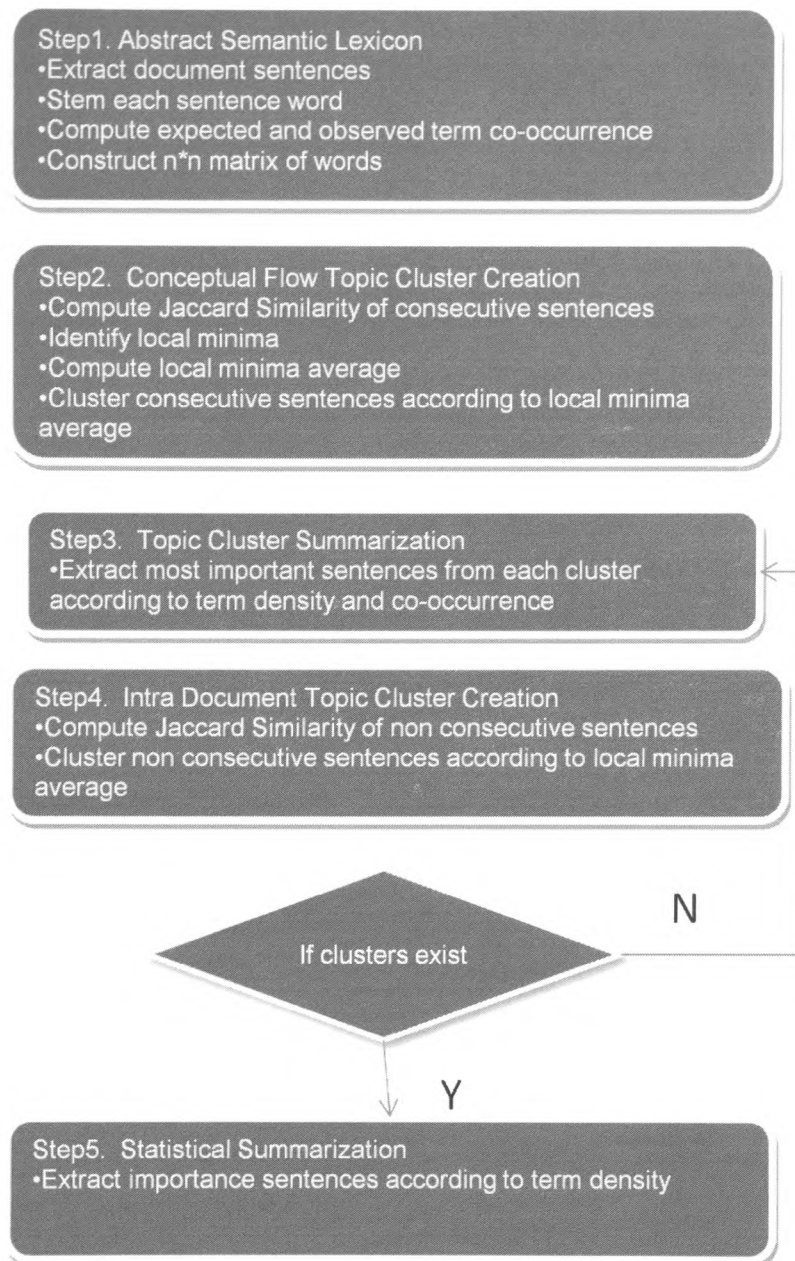


Figure 3. GUTS Summarization Phase

### 1.7 Thesis Outline

The rest of the current thesis is organized as follows:

- Chapter 2 considers statistical single-label document classification algorithms such as Naïve Bayes Classifier and Language Models. Through the analysis, the shortcomings of each algorithm are analyzed and justification on the need to develop a new statistical algorithm is provided.
- Chapter 3 describes extensively document classification and summarization. The chapter provides insight on the fundamentals of the area, as well as the

state of the art, in single-document summarization. Critical appraisal is made to provide the fundamentals of the current research. The algorithms in Machine Learning and Natural Language Processing approaches in chronological and methodological order.

- Chapter 4 provides information on linguistic characteristics and limitations of Greek language as a fundamental problem that needs to be addressed, while also it provides reasoning on the approach incorporated in both TCASGL and GUTS systems.
- Chapter 5 addresses the problem of ML-based summarization in Greek language and how it has been tackled during this research. It presents a theoretical validation of the problem and forms the basis upon which a methodology and system (TCASGL) was developed to verify the potentiality of this approach.
- Chapter 6 presents extensively the NLP methodology and system developed during current research (GUTS), its features and especially “conceptual flow”.
- Chapter 7 describes the available evaluation schemes that have been employed in the past. In addition to that, it provides information on a number of evaluation conferences that have taken or take place around the world on document summarization. The chapter concludes with the evaluation of the research systems against some baseline summarizers and other systems.
- Chapter 8 concludes this research with a critical analysis on how the aim of research has been accomplished through the objectives and the presented methodologies. Moreover, the limitations of the current approaches are presented, while potential extensions and modifications are included as to extend the functionality and improve the performance of each system.

## 2. Related Work on Statistical Document Classification

*Chapter 2 provides an overview of the related work in document classification algorithms with a special focus on statistical algorithms on Naïve Bayes Classifier and Language Models. The chapter describes both these algorithms and points out their shortcomings. This chapter also provides justification on why statistical algorithm were chosen with regards to document classification on TCASGL.*

---

### 2.1 Introduction

Single label classifiers have been extensively researched and used in areas such as spam mail identification and decision making, mostly due to their ease of use, simplicity and efficiency. Two of the most important statistical algorithms for classification are Naive Bayes Classifier and statistical Language Models. Both of these algorithms try to extract statistically important information from a document, and through a training process try to fit a random document to one of the accepted categories. Statistical algorithms were selected as they have been proven easy to implement and extremely efficient when compared to more complex algorithms on a speed-efficiency ratio. These approaches will be used as comparative algorithms for the evaluation of TCASGL classifier module.

### 2.2 Naive Bayes Classifiers

Naive Bayes Classifiers are supervised learning classifiers, based on the Bayes theorem with strong independence assumptions on feature occurrence in a random document. This implies that the occurrence of each feature (a word in this sense) in a document contributes independently to the potential class of the document. Consider a set of classes  $C$  and a set of features  $X=(x_1, x_2, ..., x_n)$ . Classification is based on the maximization of  $P(C|X)$ .

According to Bayes Theorem

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)} \quad (\text{Eq. 1})$$

Since  $P(X)$  is the same for any given class in this example set then Eq. 1 may be transformed to

$$P(C|X) = P(X|C) * P(C) \quad (\text{Eq. 2})$$

This is analyzed to

$$P(C|X) = P(C) * P(x_1, x_2, \dots, x_n | C) \quad (\text{Eq. 3})$$

and since each feature  $x_n$  is independent in the feature set  $X$ , then the final formula for calculating the probability that a given document with a set of characteristics  $X$  belongs to a category  $C$  becomes

$$P(C|X) = P(C) * \prod_{i=1}^n P(x_i | C) \quad (\text{Eq. 4})$$

The document, therefore, belongs to the class which maximizes this *a posteriori* probability, often referred to as Maximum a posteriori decision rule (MAP).

NBCs have been used extensively in document classification. They have been considered either as baseline classifiers for classification evaluation, or through extensions to the initial representation of the algorithm, with the aim to tackle known problems of the classifier. Extensions to NBC have been proposed by a number of researchers. McCallum and Nigam, for example, (McCallum & Nigam, 1998), (Nigam, McCallum, Thrun, & Mitchell, 2000) and (Dai, Xue, Yang, & Yu, 2007) proposed several extensions to NBC with Expectation – Maximization (EM) algorithms, in order to address the costly task of manually labelling an example corpus, by using a small set of labelled corpus and a large set of unlabeled corpus. More specifically, McCallum et al considered the problem of document classification using a reverse manner. Instead of calculating how a document can adapt into a category, which is inherent in Naïve Bayes, they adapted the problem to what is the probability of a class to have generated the test document. Moreover they consider the problem of classification as a set of multiple Bernoulli experiments. The input in this

case is the word and the estimation is whether a class word exists or not in the test document. However both approaches, while extending the efficiency of NBC, they suffer from the inherent problems it has, described in Section 2.2.1. Dai et al, on the other hand, considered an Expectation-Maximization approach to implement an adaptation mechanism that would eliminate the differences between the training set and the test set. Their approach performed quite well; however, it was only tested on binary classification problems, implying a strict categorization scheme, while in the initial objective set in this research a more flexible approach is required. Simply stated, the fact that none of these approaches have been developed with document summarization in mind means that they can only be used within the context of this research as an advanced approach to Naïve Bayes, rather than a meaningful application for the problem at hand.

### **2.2.1 Naive Bayes Classifier Efficiency**

The efficiency of the classifier have been outlined in (Rish, 2001) where the author proved through simulation that NBC performs best either on problems with completely independent features, which is expected given the initial hypothesis, or in cases with strongly functionally dependent features. This work also underscores the fact that the algorithm's efficiency is lower in cases with weakly dependent features. Moreover, the authors in (Rennie, Shih, & Karger, 2003) tried to find inherent problems of Naive Bayes Classifiers and correct them in order to achieve better results. Thus, they found that NBC is bias-prone if the training sets used are uneven.

This shortcoming implies that NBC cannot be used in TCASGL, since the available training datasets are not comprised by equal in number or length datasets. Moreover, a second problem is that NBC is very restrictive on label selection as it is oriented on single label classification. However, especially in document classification there are cases where the document topics might fit into more than one category, e.g. almost 1/10 of the test dataset used in evaluation consisted of documents belonging into more than one category.

### 2.3 Language Models

Another statistical approach extensively used in text classification is Language Models (LM). LM (Peng, Schuurmans, & Wang, 2004), (Ponte & Croft, 1998) are based on word co-occurrence. They evaluate this co-occurrence by assigning a probability to a sequence of words, by computing its probability distribution. When referring to document classification, a language model is associated with a document in an example set and the random document is evaluated according to the similarity with the language model. Due to the fact that it is not always possible to evaluate the language model in text corpora due to the large number of words that may constitute the language model, an n-gram approach may be followed. In an n-gram language model, the probability of the observation of a sentence  $W=(w_1, w_2, \dots, w_k)$  can be calculated as

$$P(W)=P(w_1, w_2, \dots, w_k)= \prod_{i=1}^k P(w_i | w_1, w_2, \dots, w_{i-1}) \approx \prod_{i=1}^k P(w_i | w_{i-(n-1)} \dots w_{i-1}) \quad (\text{Eq. 5})$$

Since, it is considered that the probability of the occurrence of word  $i$  of the sentence in the context history of the preceding words can be approximated by the probability of observing it in the previous  $n-1$  words. Language models have been used as an alternate approach to NBC in an attempt to evaluate the statistical dependence of words that may be apparent in a sentence. An estimation of the maximum likelihood estimate of the n-gram probabilities may be given by the observed frequency

$$P(w_i | w_{i-(n-1)} \dots w_{i-1}) = \frac{(w_{i-(n-1)} \dots w_i)}{(w_{i-(n-1)} \dots w_{i-1})} \quad (\text{Eq. 6})$$

The simplest form of Language model is the unigram language model, where all conditional information is disregarded and each term is considered independently. A unigram model according to (Eq. 7) is:

$$P_{n=1}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i) \quad (\text{Eq. 7})$$

Other commonly used language models engage bigrams (Eq. 8) and trigrams (Eq. 9), in an attempt to not only evaluate the existence of words in a document inferring

dependency between their appearance but also to evaluate the order of their appearance in the document as critical to identifying the context of a random document. Thus, “San Francisco” as a bigram that may be commonly found in random documents will be evaluated according to the dependent probability of “San” preceding “Francisco”.

$$P_{n=2}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i | w_{i-1}) \quad (\text{Eq. 8})$$

$$P_{n=3}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i | w_{i-2}, w_{i-1}) \quad (\text{Eq. 9})$$

One of the first research works in LM for Information Retrieval (IR) was undertaken by Ponte and Croft (1998), where they proposed a language modelling technique for classification tasks, and carried out experiments that proved that Language Models produced better results than standard tf.idf weighting techniques. Examples of work in language models have been undertaken (Peng, Schuurmans, & Wang, 2004) to enhance Naive Bayes Classifier with Language Model characteristics, in order to overcome the statistical independence of NBC. A direct link between NBC and LM has been observed by the authors vividly stating that unigram LM classifier with Laplace smoothing actually corresponds to the traditional NBC. In addition, the authors experimentally prove that bigram classification performs better than NBC. LMs infer ordered sequence of words as they appear in a sentence, in order to estimate the statistical dependence of the word sequence occurrence. LM have been used as a means to estimate unordered word occurrence (Srikanth & Srihari, 2002), where they proposed three approaches on estimating unordered word occurrence (referred to as biterm as it consists of two words) on random documents: through the average of the components of a bigram LM, the term frequencies of both words to the occurrence of the first observed word, and the term frequencies of both words to the minimum of the term frequencies. Their experiments showed that LM may fail to be as effective as unordered word n-ples observation.

The aforementioned research attempts lead to the conclusion that essentially LMs try to overcome the statistical independence of NBC, by introducing word sequence



dependence in the form of n-grams. This implies that they promote co-occurrence of terms in a specific order, while in reality this is not the case for a small portion of significant word sequences e.g. prime minister.

### **2.4 Synopsis**

In this Chapter, document classification algorithms based on Naïve Bayes assumption were presented. Naïve Bayes assumption provided the fundamentals of the classification approach presented of TCASGL and both NBC and LM were used as a testbed against TCASGL classification evaluation. The next Chapter introduces the idea of generic document summarization and extensively describes related work.

### 3. Related Work on Document Summarization

*This Chapter discusses extensively the background on document summarization, with focus on single-document summarization. Initially, a definition of document summarization is provided with regards to the available techniques, while an extensive analysis on basic and fundamental approaches as well as the state of the art is presented. The chapter concludes with the a brief introduction on multi-document and query-based summarization as available summarization areas that could be investigated for appropriate techniques and methodologies, as well as considered as potential extensions to the methodologies presented in this research thesis.*

---

#### 3.1 Document Summarization in General

Document Summarization is not a process that may be solved using a universal approach. Instead of a general summarization approach, a number of different techniques exist, with regards to the number of sources from which a summary may be produced, the information required, and the general manner with which the summary is presented to the user. Thus, summarization can be extracted from one source (single-document) or a number of sources (multi-document), it can cover a number of document topics (generic) or answer a specific question (query-based), it can use unaltered document sentences (extractive) or generate new sentences (abstractive) and it can use a knowledge-base(knowledge-rich) or not (knowledge-lean) use a knowledge-base.

#### 3.2 Early Generic Single-Document Summarization

Generic single-document summarization refers to the extraction or generation of a summary from a single source (document), with the aim of presenting information of the most important topics that are discussed in the document. This area of summarization was the first to be researched, with work dating back to 1950's. Thus,

a great number of approaches that apply to other document summarization areas have been initially proposed as generic single-document summarization algorithms. Apart from the historical aspects of single-document summarization, its importance has been outlined in several international conferences such as Document Understanding Conferences (DUC). In the following sections influential single document summarization techniques are described.

### 3.2.1 Statistical Approaches

Initial research in the area of document summarization exploited shallow document characteristics, through a simplistic approach of identifying document features as *term frequency* (Luhn, 1958), *sentence position* (Baxendale, 1958), *title words*, or *keyphrases* (Edmundson, 1969). The identification and extraction of important sentences is performed through a linear combination of scoring on each feature taken into account. The major concept behind each of these approaches is to identify salient sentences.

Saliency implies the existence of characteristic sentence features that promote the significance of the sentence. Luhn's (1958) pioneering approach exploited term frequency and term distribution. According to the author, highly frequent terms in a document denote its importance. Moreover, he states that important sentences feature dense key terms. He empirically identified that a bracket of 4 to 5 unimportant words between significant words is considered efficient to promote a sentence as important. However, this simplistic approach suffers from the fact that term frequency by itself can not be considered an effective feature. Other features that have been identified in bibliography include title words. As stated by (Baxendale, 1958) in his pioneering work, important topic information is expected to be found in other areas such as the document's title or in keywords (at the time user provided). Extensively researched features, provided by Baxendale, that varied significantly from term existence or frequency, were keyphrases and sentence position. According to Baxendale, the existence of word patterns denotes that the information presented before or after them is of high significance. Sentence position, on the other hand, exploits the writing style in a document, considering that the first and the last paragraph sentence usually

describe its topic. This approach alone cannot yield significant results, since it is an empiric approach not always efficient. However, it has been proven that in conjunction with other approaches, or within the context of specific domains (newspaper articles) it can actually outperform more complex algorithms. These simplistic approaches have been the basis for a number of extensions.

Document summarization tasks can be discriminated into three progressive tasks as it has been suggested by (Lin & Hovy, 2000). The first step is the identification of document topics. This enables the second step; the identification of which sentences correspond to each topic leading to a tree structure of topics, from which the final summary will be extracted. The final step is the evaluation of the significance of each sentence in the topic structure. The aforementioned features inherently try to approximate all three steps using appropriate weighting schemes.

A well known and extensively reviewed term frequency metric is *tf.idf* (term frequency to inverse document frequency) (Sparck Jones, 1972). *Tf.idf* is an empirical metric, preconditioning the existence of a training corpus. The main assumption of *tf.idf* is that term frequency can be distinctive of the topic of a document, if and only if the considered term exists scarcely in general. The general existence of the term is measured in a corpus of documents. If the term exists frequently in the documents comprising the corpus, then it is safe to consider that it represents a more generic meaning. The main form of *tf.idf* is computed according to (Eq. 10)

$$tf.idf = \frac{|word|}{\sum_i |word_i|} * \log \frac{1 + train_{doc}}{1 + train_{doc,word}} \quad (\text{Eq. 10})$$

where  $|word|$  is the number of times the word occurs in the random document,  $train_{doc}$  the number of documents in the training corpus and  $train_{doc,word}$  the number of documents featuring word. The higher the number of the corpus documents that feature word the closer to 0 is the *idf* part of the equation, and therefore the lesser the overall importance of word. This is just one form of the *tf.idf* as it has been found in the bibliography, as researchers try out different adaptations on the *idf* part of the metric. Moreover, it is important to note, that *tf.idf* is rarely used as the immediate

decision feature, since most researchers use more complex techniques in conjunction with *tf.idf*. Research using *tf.idf* in conjunction with a simple position weighting scheme has been proposed (Seki, 2002). The proposed system is based on *tf.idf* metric and a simple Baxendale-like weighting scheme on sentence position and title words that performed both single and multidocument summarization. The *tf.idf* system was trained with 230,000 newspaper articles. The author's evaluation showed that the system performed excellently only in few document from the test dataset, achieving average ratings through human evaluation. However, the author does not provide any insight on how other competing systems performed, while he did comment on the fact that sentence position is quite critical in newspaper articles and more generally heavily dependent on the document genre. This has been experimentally validated (Nenkova, 2005), where the author evaluated the findings from the DUC 2004, and explained the fact that none of the systems presented in DUC 2004 could outperform a baseline summarizer, which only considered the first  $x$  words of a newspaper article. As a matter of fact, this was the reason that single-document summarization tasks were dropped from DUC conferences thereafter.

Other systems that have utilized *tf.idf* as a weighting metric have been proposed by (Nobata, Sekine, Uchimoto, & Isahara, 2002). The authors used *tf.idf* as one of the sentence selection features. The other features used were sentence position, sentence length and headline terms. The authors applied a score function through a linear combination of each perspective score automatically adapted from a test run on the test dataset provided in NTCIR-TSC-2001 for 20% and 40% compression rates normalized so as the sum of the parameters for all four features to be 1. An interesting result the authors found was that for a 20% compression rate, the feature that contributed most was *tf.idf*, whereas for a 40% compression rate a higher sentence length contribution parameter produced the best results. This can be expected on the basis that *tf.idf* identifies words that only appear within a specific context of documents, resembling a terminology extraction mechanism. On smaller document summaries, it is more likely to identify and isolate the sentences that hold the

terminology of the document – or in this sense the topic of the document. In larger summaries, however, it is more important to include as much information as possible.

Another term weighting measure that has been proposed (Banko, Mittal, Kantrowicz, & Goldstein, 1999) and patented (Kantrowicz, 1992) is *tl.tf* (term length to term frequency). *Tl.tf* is also an empirical metric based on the assumption that word length can be indicative of word significance, as more generic words tend to be constructed by less characters than more exquisite or topic specific words. *Tl.tf* can be computed using (Eq. 11)

$$tl.tf = len(word) * \frac{|word|}{\sum_i |word_i|} \quad (\text{Eq. 11})$$

where  $len(word)$  the number of characters comprising word. *Tl.tf* has been used in research undertaken by (Kruengkrai & Jaruskulchai, 2003). (Kruengkrai & Jaruskulchai, 2003) considered a Luhn-like approach in deciding important sentences. As (Luhn, 1958), they utilized a sentence bracket consisting of both significant and insignificant words. Word significance is decided using *tl.tf* metric. This is used as a local sentence property. It is used in conjunction with a global sentence property which is decided by comparing sentences using the cosine similarity algorithm. The formula that decides on sentence significance is calculated as

$$Score = \lambda * G' + (1 - \lambda) * L' \quad (\text{Eq. 12})$$

where  $G'$  denotes the normalized global sentence property and  $L'$  the normalized local sentence property. The *Score* falls within the bracket  $[0, 1]$ . This score identifies the importance of the sentence terms, as well as how important the sentence is in the rest of the document.

Apart from text frequency, a commonly used feature that yields very good results is sentence position. Research in the area has shown that despite its simplicity it can produce very good results, to the extent that it has been used as the baseline selection algorithm in several DUC conferences. Another algorithm that has been influenced by sentence position and exploits spatial characteristics of the document has been

proposed (Strzalkowski, Wang, & Wise, 1998) on a paragraph-position level. According to (Strzalkowski, Wang, & Wise, 1998), one of the main drawbacks of sentence - based summarization is coherence, as the extracted sentences may be parts of different paragraphs. According to the authors, the sentences may lack in coherence or depict conceptual diversities. Thus, they proposed that instead of picking a sentence by its relative position in a paragraph, to consider the whole paragraph. In order to achieve this, they introduced the idea of Discourse Macro Structures (DMS). A DMS holds the rules of paragraph extraction, e.g. Introduction-Methodology-Results-Discussion-Conclusion may well be a scientific DMS, as it approaches the manner with which scientific articles are written. The extraction of the most significant paragraph is achieved through a linear combination of minor scores, instantiated differently per DMS, achieved through supervised and unsupervised approaches. The authors consider a number of significant features for news articles as: *tf.idf*, *document title*, *noun phrases*, word and phrase distribution through a metric resembling *tf.idf* called *tpf.ipf* (term paragraph frequency - inverse passage frequency), paragraph positions and cue phrases. Respectively, the features used for the background section of an article are: anaphors, dates and verbs that signify background information and the existence of proper names. Obviously, a paragraph approach targets on coherence rather than topic identification. The assumption that a paragraph may only discuss one topic, while it might be applicable in larger documents, as the scientific articles, does not imply that it is applicable in short online news articles. This happens because news articles try to provide as much information in a relatively dense form.

### 3.2.2 Machine Learning Approaches

Statistical approaches. Machine Learning (ML) approaches in computational linguistics have been extensively used in areas such as document classification. Famous algorithms such as Naive Bayes Classifier and Language Models have been proposed and reviewed in document classification, but little research had been accomplished in exploiting ML in document summarization. Machine Learning approaches are always knowledge rich approaches, since they require a training set in order for the system to perform summarization tasks. Initial work in the area was

performed by (Kupiec, Pedersen, & Chen, 1995). The authors identified a series of features, following Baxendale's work, but instead of using a linear combination of the features' weight, they proposed a simple Bayesian classifier that was trained to estimate the significance of each sentence per discrete feature. The sentence score was assigned according to the probability score of each feature as calculated by the classifier. The sentence score denoted the probability of whether a random sentence should be included in the final summary or not. However, their evaluation proves that regardless of the features they selected, their approach did not yield significant results. Moreover, the authors fail to compare their system with any other system. A similar approach has been proposed (Aone, Okurowski, Gorinsky, & Larsen, 1997). The authors also proposed a Naive Bayes Classifier but utilized richer features. More specifically, they utilized *tf.idf* metric in identifying keywords in a similar manner to the one described in (Eq. 10). The authors incorporated shallow discourse analysis and semantics in identifying term similarity as e.g. USA: United States of America. This assisted in approximating word synonymy and morphological variants, increasing the efficiency of the *tf.idf* metric. Another approach has been considered by (Nomoto & Matsumoto, 2001). In their work the authors concentrated on two features, with the aim of preserving and representing the full topics covered in the document. In order to achieve the document topic representation, they adapted the K-means clustering algorithm so as to automatically consider both the initial points upon which the clustering will occur and utilize Minimum Description Length Principle to achieve the best possible cluster modelling. This initiated a cluster model through diversity identification by eliminating distortion (the initial X-means points were decided upon the average Euclidian distances of each cluster), while the summary was produced by the best scoring sentence of each cluster. The score formula was decided by a modified *tf.idf* metric on each cluster.

A similar approach has been presented by (Wang, Li, & Wang, 2007). The authors propose a similar three step approach as the one proposed in (Nomoto & Matsumoto, 2001). However, these methods, vary slightly in both estimating term significance and deciding on the thematic sentence of each cluster, as (Wang, Li, & Wang, 2007)



utilize *tl.tf* for word significance instead of *tf.idf*, and thematic sentence is decided through the cosine similarity of the centroid of each cluster, rather than the *tf.idf* score proposed by Nomoto and Matsumoto. Results, however, cannot be comparable since summarization tasks are heavily dependent on the evaluation test corpus, and the fact that Nomoto and Matsumoto's approach is supervised, while Wang, Li, and Wang's is unsupervised.

Hidden Markov Models. Research in the area of HMMs for use with generic summarization tasks was initially undertaken by (Conroy & O'Leary, 2001). The authors proposed a series of feature sets as sentence position, number of terms, number of terms existing in the training dataset and term frequency over the document. Since sentence position in the paragraph is a measure that is dependent on the relative position of the sentence instead of using a naive Bayesian approach, with implications of statistical independence, the authors proposed the use of an HMM. The use of a HMM also considers the variable probability of a sentence being included in the summary given that the previous sentence is actually a summary sentence. Sentence selection is decided either through maximum posterior probability calculated by the HMM, or through a QR matrix decomposition approach that eliminates topic redundancy. The QR decomposition algorithm, also described in the paper, forms a term by sentence matrix and extracts important sentences according to term occurrence. The matrix is updated by subtracting the component of each extracted sentence from the remaining columns of the matrix, as the topic the extracted sentence describes (through its terms) has been included already in the summary. The algorithms seem to perform fairly well in comparison to human summaries, even though the authors acknowledge that the algorithms could be enhanced with more features from NLP.

Position-based approaches. Important work in the area of ML in generic document summarization has been proposed by (Lin & Hovy, 1997). Using Baxendale's position feature, they solved the problem of genre diversity and how it affects topic sentences. The initial problem lies in the fact that the topic sentence, which Baxendale stated may be the first or the last sentence of a paragraph, is actually dependent on the genre

---

of the document. Thus, for different document genres, authors place the most important sentence in different positions in their paragraphs. The authors, using a set of training documents and their summaries, tried to locate the *Optimal Position Policy (OPP)* that would provide an estimate on the position of the topic sentence in a paragraph. Moreover, an assisting feature the authors used is in extending the efficiency of their system is user-provided keywords.

### 3.2.3 Natural Language Processing Approaches

Natural Language Processing (NLP) has also been utilized in document summarization. The underlying basis of these approaches lies in identifying and exploiting semantic relations of terms of the documents, identifying sentence discourse structure and evaluating sentence significance through term co-occurrence. These approaches can be knowledge rich (in cases where reference data is used in the form of semantic lexicon, document analysis or training datasets), or knowledge poor (in cases where no external anaphora is used). Research in the area of NLP revolves around two axes: Semantics and Rhetorical Structure Theory (RST).

Semantics. Semantics in Generic Single-Document Summarization have been considered both through knowledge rich approaches, where semantic lexicons such as Wordnet (Miller, 1995) are used to evaluate term relations, and knowledge poor approaches, where semantic analysis is performed on a single document with the aim of approximating word relations. (Barzilay & Elhadad, 1997), for example, performed deep linguistic analysis through the use of Wordnet. The authors proposed a feature named lexical chains, which corresponds to a sequence of semantically related words that span across text. They considered short lexical chains, spanning across adjacent words or sentences, and long lexical chains, spanning over the entire document. Their aim is to segment the text, identify the chains, and decide through heuristics on the most important of them in order to identify the most significant sentences to extract. A lexical chain is not only considered on term presence, but rather on the existence of semantically related terms across the document. Semantic relation is acquired through the semantic distance of terms within Wordnet lexicon. This contribution is very important as it set the basis the use of semantics and inferred knowledge in document

summarization. The most important disadvantage, if it can be stated as such, is the requirement for an external resource to identify word relation, and to estimate topic distribution within the document.

Approaches that avoid this problem try to estimate topic identification using document resources. A commonly used semantic method in this area, based on Singular Value Decomposition (SVD), patented in (Deerwester, et al., 1988), is Latent Semantic Analysis (LSA). LSA tries to evaluate the contribution of a word in a text segment (extending from a sentence to document in cases of multi-document summarization) as well as the importance of a text segment featuring a word. LSA succeeds in both identifying noun phrases (San Francisco) and identifying the different topics presented in a document. The first step of the algorithm is to construct a matrix of terms by sentences. Considering that, generally, document terms ( $m$ ) are unequal to document sentences ( $n$ ),  $A$  is an  $m \times n$  matrix.  $A$  is a very sparse matrix as not all terms contribute to every sentence. Through SVD, matrix  $A$  is decomposed in

$$A = U \times \Sigma \times V^T \quad (\text{Eq. 13})$$

where  $U$  is a column-orthogonal matrix holding the left singular vectors,  $\Sigma$  is a diagonal matrix whose values are sorted in descending order and  $V$  is an orthonormal matrix holding the right singular vectors. As stated in (Gong & Liu, 2001), from a transformation point of view, SVD provides a mapping between each left singular vector (word) and each right singular vector (sentence). From a semantic point of view, SVD represents the analysis of the original document into concepts, captured into the singular vector space. In addition to that, it enables the establishment of strong relations between semantically related terms, as they will be projected very close to the singular vector space, as they share a great number of common words. The authors conclude that in SVD, each singular vector denotes a salient topic, whereas the magnitude of the vector denotes the importance of the topic. As stated by the authors of LSA (Landauer & Dumais), in contrast to identifying term co-occurrence, LSA tries to estimate the average meaning of a passage (e.g. sentence, paragraph or document) from the terms it consists of.

Discourse Structure. RST has also been considered as an option to identifying the topic of a document. RST has been proposed (Mann & Thompson, 1988) and used for single document summarization tasks (Marcu, 1997). The author identified the problem that commonly used features, such as title-based keywords, may lead to the inclusion of subsidiary to the title, sentences in the summary. Through RST, the author applied a discourse analysis on the document and extracted a tree of primary and secondary meanings. These are represented as NUCLEUS and SATELLITES. The main difference according to Marcu's (1997) approach between a NUCLEUS and SATELLITE is that a NUCLEUS is a comprehensible text segment by itself as opposed to a SATELLITE. Rhetoric parsing is utilized on the document and a rhetorical structure tree is extracted, by using features as cue phrases and simple semantics. Summarization is achieved through partial ordering of the resulting tree, and the isolation of the NUCLEI closer to the main NUCLEUS. In order to identify the effectiveness of RST, Marcu utilized a series of generic features (term frequency) and specific to his case features, e.g. shape of the resulting tree. Marcu's approach is closer to the linguistic side. RST is a linguistic method that specifies the way meanings are represented in documents through a series of relations. Marcu only included parts of RST, the main problem being that the full RST definitions cannot be machine extracted. Thus, his work, while being pioneering, suffers from the fact that the result tree partially represents sentence and word relations.

Additional work in the area has been undertaken by (Paice & Jones, 1993). According to the authors, abstraction based on document structure can be trained, and important word templates can be identified. They isolated through training a series of semantic patterns that can be adapted on a specific document genre, e.g. scientific papers, on a specific domain, e.g. agriculture. However, they identified that generalizing on different domains may not be as easy, and requires an extensive training phase to accumulate semantic patterns. Their analysis on agricultural documents provided an adaptable weighing scheme per semantic role, while the selection feature is the sum of each candidate term sequence per role as they have been identified in the document. However, the efficiency of their approach relies on how well the document structure

templates are defined, while a limitation of the algorithm is the fact that it is oriented mostly on technical articles.

### 3.3 Modern Approaches in Generic Single-Document Summarization

This section presents work that has been carried out in the area of generic single-document summarization during the last decade. Considering the proposed categorization of approaches (shallow approaches, ML, and NLP), a presentation of the state of the art is made with the aim to include as many of the advancements that have been made during the last decade and evaluate the trends in document summarization. During the last decade a considerable amount of research has been carried out. Research, of course, has derived from simplistic approaches and focuses mostly on ML and NLP approaches while shallower analysis is mainly used in the initial steps of each algorithm and in conjunction with more complex techniques. Thus, in the rest of the section, focus is on ML and NLP approaches; however references to the underlying shallow technologies used is presented, when applicable.

#### 3.3.1 Machine Learning Approaches

Modern ML approaches include a broader scope of technologies than the one presented already. The most usual ML techniques nowadays use mainly Statistics or Artificial Intelligence for the evaluation of word significance.

Statistical Approaches. An approach on statistical ML approach has been proposed by (Amini & Gallinari, 2002). Following (Kupiec, Pedersen, & Chen, 1995), the authors present a logistic regression classification semi-supervised approach on document summarization. The basis of their work is the probability that a sentence will be included in a summary or not. However, instead of utilizing a Bayesian approach, they consider a training set of labelled and unlabeled documents, where labelled data assumes that each document sentence has been marked for inclusion in summary or not. Using the Classification Expectation Maximization (CEM) algorithm on the training dataset, the authors try to maximize a parametric formula of unknown parameters, defined as Classification Maximum Likelihood (CML) criterion. For their evaluation they used a query-based *tf.idf* system and Kupiec's algorithm, while their

approach outperformed both of them, in terms of *Precision* and *Recall*. It would be interesting, however, to estimate their approach against more modern qualitative metrics as ROUGE. The same authors considered a slightly different approach on ML single-document summarization utilizing instead of classification, ranking of sentences (Amini, Usunier, & Gallinari, 2005). Their research utilized a set of widely used features as *indicator phrases* and *title keywords*, which were used to score a sentence. Through expansion on title keywords, using Lexical Context Analysis, Wordnet and Word Clustering, they formed queries with which they compared every sentence set. Sentence similarity was calculated using *tf.idf* on a term and acronym level. Sentence ranking was achieved by comparing pair of sentences through which the authors try to evaluate which sentence should be included in the final summary. Each pair is correctly classified if, and only if, the score of the sentence to be included in the summary (relevant) is greater than the irrelevant. This feature was used to train a ranking formula, upon which the statistical error was computed. Summarization is achieved through a minimization of the loss function using logistic regression.

A slightly different statistical approach has been proposed by (Shen, Sun, Li, Yang, & Chen, 2007). According to the authors summarization is a process of labelling on sequential document sentences. In order to extract a summary, a human must read the document and decide whether a document sentence - that is *de facto* connected to a document topic – is important enough to be included in the summary. The process includes the utilization of a series of features that have an adaptable importance score. Their importance score is calculated sequentially, while the document is scanned, using Conditional Random Fields (CRF). According to CRF each sentence is treated as a state, for which a global label score may be assigned. The score is dependent on the previous and the current state, and is affected by a series of parameters namely: position, length, log likelihood, term frequency, indicator words, similarity to neighbour sentences, LSA scores and HITS scores. In order to compute the CRF for a sentence  $i$  the authors define the formula (Eq. 14)

$$CRF_i = \frac{e^{\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, X) - \sum_{i,l} m_l g_l(y_i, X)}}{Z^X} \quad (\text{Eq. 14})$$

where  $\lambda_k$  and  $m_l$  the weights learned at state  $i-1$  and  $i$ , and  $f_k$  and  $g_l$  the feature functions at state  $i-1$  and  $i$ . The parameter estimation is maximized through a maximization likelihood procedure on the conditional log likelihood of the labelled sequence.

Artificial Intelligence Approaches. Apart from statistical training on the importance and weight of the selection feature set, research also includes broader AI techniques. In (Jaoua & Ben Hamadou, 2003), the authors propose a classification scheme for summarization based on a Genetic Algorithms. Their proposed process includes a statistical module calculating *word and lemma frequency*, a *discourse module*, identifying rhetorical structure rules by combination of key-phrases and a generation and classification module that produces all possible extracts and classifies them. The classification is dependent on a *length indicator* (how long does the user want the summary to be), *keyword coverage* (number of keywords found in the extract compared to the number of keywords found in the original document - keeping in mind that keywords denote topics), *weight* of each sentence (based on keywords present in each sentence), a discourse similarity indicator that compares each extract to predefined templates and a cohesion indicator. The classification step determines the best possible candidate of all extracts as the summary. Such an approach, however, can only be applicable for fairly reasonable in size documents, as the more the sentences the more the number of potential extracts.

A similar methodology has been proposed by (Yeh, Ke, Yang, & Meng, 2005). The authors consider five features as *position*, *positive keywords*, *negative keywords*, *centrality* and *resemblance* to the title to train their GA. The features are considered using an adaptable linear combination on their weights. A different approach, based on Neural Networks was considered by (Svore, Vanderwende, & Burges, 2007). The authors utilized RankNet NN ranking algorithm in creating sentences that describe the highlights (most important topic) of a document. The process included evaluating the similarity of sentence pairs from a document using ROUGE (Lin & Hovy, 2003), identifying candidate sentences based on supplied highlights on the training phase, as well as re-adapting the importance of a series of features on each training step. The

features considered are: First Sentence, Position, SumBasic - a score dependent on term frequency and sentence calculated using (Eq. 15), Title Similarity and a series of features from online sources such as Microsoft News and Wikipedia entries.

$$SumBasic(S_i) = \frac{\sum_{w \in S_i} p(w)}{|S_i|} \quad (\text{Eq. 15})$$

where  $p(w)$  is the probability of term  $w$  in a document and  $|S_i|$  the number of words in sentence  $i$ .

Both Genetic Algorithms and Neural Networks have been considered fairly recently in (Fattah & Ren, 2009). The authors gathered a set of 10 statistical, position and semantic features, some of which were already defined in (Yeh, Ke, Yang, & Meng, 2005). These features were used as weighted input parameters on a GA that tried to calculate the optimum linear combination between them. as well as in a Feed Forward (FFNN) and a Probabilistic (PNN) Neural Network, that primarily aimed at classifying a sentence as a summary sentence or not, through the input features. All three approaches yielded comparable evaluation results and outperformed Yeh's et al (Yeh, Ke, Yang, & Meng, 2005) Genetic Algorithm.

Other Approaches. Another technique presented in the literature on ML single document summarization involves the training of a Support Vector Machine for sentence extraction. Support Vector Machines are generally used in classification problems, and in accordance with all ML approaches presented until now, they can be used to promote an extract as the summary. This has been researched by (Li, Zhou, Zha, & Yu, 2009). The authors researched single-document summarization forming three key qualitative characteristics: diversity, denoting less information redundancy; coverage, denoting that the information covers as many of the topics as possible; and balance, denoting the lack of overemphasizing a topic on the expense of some others. The SVM forms a discriminant feature function through a linear combination of the feature set. The feature set considers word frequency, position, thematic word, sentence length, uppercase words, PageRank and bi-grams. Given a document  $x$  the



SVM creates all possible extracts and based on a training dataset, tries to maximize the discriminant function of the corresponding weighted features.

### 3.3.2 Natural Language Processing Approaches

A very important domain of science that has been applied to generic single-document summarization with extraordinary results is NLP. Typically, modern NLP approaches consider shallow (LSA-based) or deep (Wordnet) semantic analysis, featuring semi-supervised and unsupervised approaches, as well as document analysis theory (RST). This section presents a number of NLP methodologies that have yielded significant results during the last decade.

Lexical Chains and Wordnet. Following the inspiring work by (Barzilay & Elhadad, Using Lexical chains for Text Summarization, 1997), a number of approaches featuring lexical chains have emerged. For example, (Angheluta, De Busser, & Moens, 2002) proposed a lexical chaining approach using Wordnet's synonymy relation on a noun level among other approaches to achieve topic segmentation. Their approach considered the construction of a tree-like table of content (ToC), consisting of the document topics. In order to construct the ToC, the authors used as an option lexical chaining, the identification of the topic of each sentence through heuristics as term persistence and position as well as term distribution over the document. The summary was extracted by the level of detail dictated from the desired summary length.

Another example of lexical chains has been proposed by (Song, Han, & Rim, 2004). The authors maintained the initial thoughts of (Barzilay & Elhadad, Using Lexical chains for Text Summarization, 1997), however, they extended this approach by calculating the probability that a lexical chain is correct. This was achieved by examining word co-occurrence, the depth in Wordnet hierarchy as well as the semantic relation of terms as provided by Wordnet. They formulated a product function that provided the chain score from these relations. The sentence score was calculated using a product function on word connectivity and chain connectivity, denoting word connectivity in the chain and global chain connectivity respectively.

The sentence score itself was the sum of this product per chain in each sentence. Apart from lexical chaining Wordnet has been used in general scope summarization tasks. Such tasks have been proposed by (Bellare, et al., 2004). Their approach uses Wordnet to extract the graph that is closest to the document at hand. This is estimated using a cut-off depth feature. Each resulting synset is ranked according to term presence in the document, while sentence selection is decided on the number of synsets each sentence has. Principal Component Analysis is applied to assist in acquiring most relevant sentences from the document.

Wordnet has also been used to a less extent by (Filatova & Hatzivassiloglou, 2004). The authors proposed a novel feature based on the interaction between named entities or frequent nouns, referred to as action events. Wordnet in their example was used to capture these action events, and filter out irrelevant concepts. In their system they analysed and acquired name entities/frequent noun pairs from textual units (sentences in their case) and counted their occurrence. Using Wordnet they formed triplets within each pair - the third part of the triplet denoting the action. Each action connector and name entity/frequent noun is assigned a core based on their normalized appearance. The action event score is calculated by the product of these two scores. Sentence selection is made by gathering the minimum number of sentences covering the most important concepts.

All of these approaches, while being very efficient since they utilize full semantic relation information, introduce an extra costly step. Querying Wordnet to retrieve, evaluate and use semantic information is a very “expensive” step in real-time document summarization tasks.

A semantic approach that does not utilize Wordnet has been proposed by (Saggion, Bontcheva, & Cunningham, 2003). In their approach, the authors used the GATE system to transform input documents and a set of statistical and linguistic features to perform sentence selection. Such features are: corpus statistics; *tf.idf*; cosine similarity of sentences; named entity; sentence position and input query. Sentence selection is

based on a user-supplied linear combination on these features. However, the authors provide little information on the efficiency of their approach.

Latent Semantic Analysis. LSA and SVD based approaches have been extensively researched during the last decade. As has been suggested by Steinberger and Jezek in (Steinberger & Jezek, 2004) and (Steinberger & Jezek, 2005), LSA suffers from two main disadvantages. The first is that in LSA one must pick the  $r$  value in the singular vector space ( $\Sigma$  matrix) according to the number of sentences of the summary. Since each singular value denotes a distinctive topic or subtopic, this is helpful only if one knows the subtopics of the documents. The second drawback the authors identified, is that if a sentence has large index values, but not the largest in any dimension, it will not be chosen, despite the fact that it may contain important information. In order to tackle these deficiencies, the authors propose a modification of LSA. In (Eq. 13) instead of considering  $V^T$ , the authors computed the length of each sentence vector according to (Eq. 16)

$$s_k = \sqrt{\sum_{i=1}^n v_{k,i}^2 * \sigma_i^2} \quad (\text{Eq. 16})$$

where  $s_k$  is the length of the vector of sentence  $k$  in the latent semantic space,  $v_{k,i}$  is the value of the vector right singular vector in the  $(k,i)$  point, and  $\sigma_i$  is the value in the  $(i,i)$  singular vector space.  $n$  in this example is the dimension of resulting vector. As it may be observed, this approach tackles efficiently both problems described, as the proposed methodology is independent on the length of the summary, while the product of each sentence value with its corresponding singular value, assures that sentences with a general high index will be included in the final extract. Indeed this was verified in the evaluation of the adaptation, as the authors' methodology surpassed (Gong & Liu, 2001) approach.

An adaptation on LSA has also been proposed by (Yeh, Ke, Yang, & Meng, 2005). The authors, instead of following the straightforward methodology of using term frequency per matrix, defined a score function dependent on both local and global characteristics. The function used is (Eq. 17)

$$a_{ij} = G_i * L_{ij} \text{ (Eq. 17)}$$

where  $a_{ij}$  the coefficient of term by sentence matrix  $A$  in (Eq. 13), where  $L_{ij}$  the local score of term  $i$  in sentence  $j$ , while  $G_i$  the global score of term  $i$  in the document.  $L_{ij}$  is computed by (Eq. 18)

$$L_{i,j} = \log(1 + \frac{c_{i,j}}{n_j}) \text{ (Eq. 18)}$$

where  $c_{ij}$  the frequency of term  $i$  in sentence  $j$  and  $n_j$  the number of words sentence  $j$  has. The global score  $G_i$  is computed by (Eq. 19)

$$G_i = 1 - E_i \text{ (Eq. 19)}$$

where  $E_i$  is the normalized entropy of term  $i$ , computed by (Eq. 20)

$$E_i = \frac{1}{N} \sum_{j=1}^N f_{i,j} * \log(f_{i,j}) \text{ (Eq. 20)}$$

where  $N$  the number of sentences in the document and  $f_{ij}$  the frequency of term  $i$  in sentence  $j$ . By introducing the entropy feature the authors manage to give a probabilistic basis to LSA, and thus to allow a smoother evaluation of the topic distribution.

An extension to LSA has been proposed by (Bhandari, Shimbo, Ito, & Matsumoto, 2008). The authors claim that one of the drawbacks of LSA is its lack of statistical grounding. In order to tackle this disadvantage, they engage Probabilistic Latent Semantic Indexing in document summarization (PLSI). PLSI is used for optimization problems, and treating summarization as such, the authors try to statistically evaluate the the maximum likelihood principle for each word-sentence pair to belong to  $z$ , given a set of sentences  $s \in S : [s_1, s_2, \dots, s_n]$ , a set of classes  $z \in Z : [z_1, z_2, \dots, z_n]$  denoting the document topics and the set of document words  $w \in W : [w_1, w_2, \dots, w_n]$ . The likelihood principle is defined as  $P(d, w) = P(d) \prod_z P(w|z)P(z|d)$ . The maximum likelihood principle is acquired by maximizing the log-likelihood function  $L = \sum_d \sum_w n(d, w) \log P(d, w)$ . Optimization is achieved using Expectation- Maximization.

Their evaluation results were found to significantly outperform standard LSA (Gong & Liu, 2001) approach.

A different approach to LSA, sharing a similar basis is proposed by (Lee, Park, Ahn, & Kim, 2009). According to the authors, another disadvantage of LSA is the fact that it generates negative vectors in the left and right eigen-vectors. In essence, this implies that words contribute negatively to a topic in the document. Therefore, eliminating these negative eigen-vectors should adapt more closely to the topics of the document. According to the authors, this is tackled by a different analysis on the term by sentence matrix named Non-Negative Matrix Factorization (NMF). NMF is applied on a term by sentence matrix which is decomposed into two non-negative matrices using the Frobenius norm. Their results outperform the standard LSA algorithm proposed by (Gong & Liu, 2001).

Rhetorical Structure Theory and Discourse Analysis. Apart from deep or shallow semantic analysis, NLP techniques exploiting the document structure have been proposed. These techniques try to identify important lexical or structural information in the document.

For example, (Moens, Angheluta, & Dumortier, 2005) propose an approach similar to Marcu's (1997) approach. The initial step in their single document summarization is to clean up the document using text removal and linguistic analysis. The second step in their analysis includes two main operations: detection of the main topic of a sentence term distribution and hierarchical table of content extraction. In the first step, the authors try to detect the main topic discussed in a sentence by considering the initial position of a noun phrase and its persistence across consecutive sentence. The second step is an analysis on the term distribution which provides important information regarding term frequency, co-occurrence and term proximity. This step examines topic shifts as well as the identification of nested topics (sentences that may discuss more than one topic). Thus, a hierarchical table of content is produced, featuring all the topics discussed in the document. Deciding on the level of detail required enables the selection of the sentences from the table of contents. This, of

course is major improvement over Marcu's approach, and to some extent (topic shift and regression identification) influenced GUTS.

Graph-Based Ranking Approaches. The theory of graphs for ranking has been widely proposed in single document summarization. The main idea is to rank sentences according to their saliency and extract sentences from the most salient ones. (Mihalcea & Tarau, 2004), for example, proposed a graph-based approach, using a modified version PageRank algorithm, named TextRank. TextRank analyzes the text into sentences and constructs a weighted graph, the vertices being the sentences and the weighted edges the similarity between the sentences. TextRank follows a standard ranking scheme computed by (Eq. 21)

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (\text{Eq. 21})$$

where  $d$  a damping parameter between  $[0-1]$ , typically set at 0,85,  $V_i$  the vertex (sentence)  $i$ ,  $In(V_i)$  the vertices linking to  $V_i$  and  $Out(V_i)$  the sentences  $V_i$  links to. In order to compute the weight  $w_{ij}$  between two sentences formula (Eq. 22) is used.

$$w_{ij} = \frac{S_i \cap S_j}{\log(S_i) + \log(S_j)} \quad (\text{Eq. 22})$$

where  $S_{i,j}$  sentences  $i$  and  $j$ . TextRank algorithms performs very well, although it seems to perform better in multi-document summarization.

On the other hand, (Zha, 2002), defined and analyzed the mutual reinforcement principle as his approach on document summarization. According to (Zha, 2002), a term is assigned a high saliency score if it is present in many sentences with high saliency scores, while a sentence is assigned a high saliency score if it contains a great number of terms with high saliency scores. This enables the creation of a bipartite weighted graph between terms that exist in sentences. This idea is also present in (Wang, Li, & Wang, 2007). However, the authors instead of considering only Sentence-Word Relations as Zha (2002) extended the reinforcement principle to a

Word-Word and Sentence-Sentence level. Their idea was found to outperform both Mihalcea and Tarau's (2004) and Zha's (2002).

### 3.4 Query-Based and Multidocument Summarization

While this thesis is targeted on generic single-document summarization, in this section summarization methodologies will be briefly discussed along with several examples.

Query-based summarization in contrast to generic summarization tries to cover a specific topic discussed in one or more documents, rather than cover as many topics as possible. There is little difference between the technologies that can be used; however in query-base summarization it is important to filter-out unwanted information. An example of query-based summarization has been proposed by (Bosma, 2005). In his research the author uses Rhetorical Structure Theory (RST) (See Chapter 3.2.3) to construct a document graph. This document graph is weighted according to the number of discourse units identified (NUCLEI and SATELLITES), the number of sentences in each SATELLITE and the number of words in a sentence. Thus, each sentence is weighted against the relevant - to the initial user query- meanings depicted (through weighting NUCLEI), while also providing additional relevant information (SATELLITES).

Query based summarization has also been considered by (Saggion, Bontcheva, & Cunningham, 2003). The authors use a generic summarization module with extensions for query-based summarization. This is accomplished by a linear combination of a number of features, one of which (*query-based scorer*) is specific to the query-based summarization task. *Query-based scorer* is a feature that tries to identify the similarity of a sentence a user query through term comparison.

Apart from query-based summarization, a lot of research has been undertaken in the area of multidocument summarization. Multidocument summarization refers to the extraction of a summary through multiple sources that deal with the same topic. Multidocument summarization provides a number or surplus limitations that are taken into consideration as: topic redundancy, time sequence and cohesion. These features, while apparent in single-document summarization, do not play a very important role, since

in a single-document it is generally recognised that the author maintains a specific authorship style, does not make time shifts, and generally decide from which aspect of the story they will refer to. Multi-document summarization systems, however, must ensure that redundancy is avoided, that information is presented without time hops, and that cohesion is maintained, which may not always be an easy task when the information is gathered from multiple sources. Multi-document summarization is an area of interest where a lot of research is being carried out, especially since the drop of single-document newspaper summarization tasks from DUC conferences in 2004. In the next few paragraphs, the most influential work, as well as some modern approaches, on multi-document summarization is presented.

Pioneering work in the area of multi-document summarization has been undertaken by (Radev, Jing, & Budzikowska, 2000) that resulted in MEAD summarization toolkit. In this work a centroid based approach was considered. The authors consider a centroid to be a pseudo-document of words within a cluster on a specific subject. Each sentence is checked against its similarity to the centroid. Using a linear combination of parameters as the centroid score (calculated using *tf.idf*), the position of the sentence in a document, and the frequency of terms (using *tf.idf*), MEAD system extracts the most significant sentences which are ordered according to chronological characteristics.

Centroids have also been considered by (Radev, Fan, & Zhang, 2001). The authors utilize a similar approach to MEAD system, offering a personalised mode, where the users set their own weight formula in retrieving information that suits them best.

A similar approach based on the main ideas behind MEAD system has been proposed by Erkan and Radev (Erkan & Radev, 2004a). Sentence centrality to the cluster is computed using a new feature the authors refer to as *Prestige*. *Prestige* is calculated using the PageRank algorithm and shows the relation between each sentence. This is calculated using a graph representation of the sentences discussing a common topic. Sentence similarity is calculated by an idf-modified version of the cosine similarity



metric, adapted to evaluate on the importance of sentence terms, according to their overall occurrence over the multidocument data set (Erkan & Radev, 2004b).

An equally important approach in multi-document summarization has been proposed by (Carbonell & Goldstein, 1998) and (Goldstein, Mittal, Carbonell, & Kantrowicz, 2000). Instead of using centroids, the authors define Maximal Marginal Relevance (MMR) as the linear combination of features as *novelty* and *relevance*: novelty denoting the diversity between two given documents; and relevance, the similarity between a document and a query or another document. This linear combination is affected by a variable which is used to assign a greater weight to each one of these features. Sentences that maximize this Marginal Relevance are extracted for inclusion in the summary.

MMR has been considered as an option by (Lin & Hovy, 2002). Following their work in OPP (Lin & Hovy, 1997), they proposed a similar approach using MMR in order to identify and evaluate sentence pairs for sentence reduction, OPP in order to extract the positional characteristics of sentence importance and the Webclopedia (Hovy, Gerber, Hermjakob, Junk, & Lin, 2000) ranking algorithm to rank the sentences by context.

Following the same basis of MMR, work on the information richness and novelty has been undertaken by (Wan, Yang, & Xiao, 2007). In their work, the authors consider query-based multidocument summarization through *Manifold-Ranking*. *Manifold-Ranking* is the process of identifying biased information richness (relation of a set of sentences to a topic) and information novelty (diversity of sentence in the summary topic), combined linearly. Topic similarity is identified using cosine similarity measure, while novelty is identified using  $tf*idf$  (modified  $tf.idf$  on a sentence level)

A graph-based multi-document approach has also been considered (Mani & Bloedorn, 1997). In this approach the authors construct a graph, where each node denotes a word and each vertex the weight of connection between words. This is calculated  $tf.idf$  over a training corpus. The extraction of the summary uses phrases as the basic textual unit, which is constructed using templates of words. Text is extracted by trying to cover as

much from the information included in the document sets, minimizing textual redundancy.

Single-document techniques have also been extended for multidocument summarization. In Columbia summarization engine (McKeown, Barzilay, Evans, Hatzivassiloglou, Schiffman, & Teufel, 2001), for example, proposed a system comprised of a system that identified and classified a set of documents in four categories: single-event documents, multi-event documents, biographies and other. The authors proposed two systems that could perform multi-document summarization according to the document-genre the system identified. The first sub-system, named MultiGen, utilized sentence level evaluation on similarity, along with positional characteristics, to calculate sentence importance. Sentence extraction was accomplished using lexical chains (Barzilay & Elhadad, 1997). The second sub-system named DEMS utilized Wordnet, named entities sentence position, and presence of pronouns to calculate sentence similarity. An interesting fact is that contrary to what had been considered until then, the authors proposed the use of verbs as the main lexical significance measure, identifying how commonly a verb may be used in documents.

Another example of single-document techniques that have been migrated to multi-document summarization is offered by (Conroy, Schlesinger, Goldstein, & O'Leary, 2004), where extensions to their HMM model (Conroy & O'Leary, 2001) for multi-document summarization was considered.

### **3.5 Synopsis**

In this Chapter, related work on generic single document summarization is presented. The presentation focuses both on ML and on NLP approaches, while query-based and multi-document summarization approaches are also presented to a smaller extent. The next chapter provides insight on the grammatical and syntactical particularities of the Greek language, as well as validation on a series of limitations and how they can be dealt with in developing TCASGL and GUTS.

## 4. Linguistic Analysis for TCASGL and GUTS

*Chapter 4 provides an analysis on the linguistic characteristics that apply both to TCASGL and GUTS algorithms. The main feature considered in this Chapter is the use of nouns and adjectives as the main word significance feature in both systems and the reasons why they were considered as the main structural units in the topic extraction of both systems.*

---

### 4.1 Nouns, Adjectives and Their Importance

One of the main problems that both ML and NLP problems have to face, regarding the extraction of the topic of a sentence, is the identification of the important, from a topic-wise aspect point of view, words. As claimed in (Bouras & Tsogkas, 2010), nouns and noun phrases are considered to be of primary importance in such a task, as they hold information on who or what acted and who or what accepted the outcome of this action. Thus, it is crucial for the efficiency of both TCASGL and GUTS, to identify the nouns of the sentence, rather than other grammatical elements. In the case of both systems, an extension to the proposed methodology was taken, by considering the adjectives, as well. Adjectives are used to characterize nouns or noun phrases, to the extent that certain nouns are almost always accompanied by certain adjectives. Therefore, semantic information can be extracted both from the nouns of a sentence, and from the accompanying adjectives. The extraction of such grammatical elements can be achieved either by applying a syntactical analysis on each of the sentences using a Part-Of-Speech (POS) tagger as in (Moens, Angheluta, & Dumortier, 2005), or applying a grammatical analysis as in the algorithms presented in this research.

### 4.2 Syntactical Analysis – Part of Speech Tagging

POS Tagging is an effective way of identifying the parts of speech apparent in a sentence; namely subject, verb object or predicate, as well as explanatory sub-

sentences or words. The main linguistic characteristic exploited in such a case is the Subject-Verb-Object (SVO) sentence order followed in most Indo-European languages. A syntactical analysis is performed to identify the verb of the sentence which provides efficient information on the subject and the object of the sentence (it being either a noun or a whole sub-sentence along with the adjectives or adverbs that may constitute a noun phrase). Part of speech tagging, however, is generally prone to errors arising from grammatical or syntactical deviations of the reference language. Firstly, the extraction of a noun or noun phrase, serving as the subject or the object of a sentence, does not imply that it may be found in this exact form over a number of random documents, apart from cases where these are names (e.g. Mona Lisa, Mount Everest etc). This phenomenon is intensified in languages where grammatical deviation is applied on the whole noun phrase. Secondly, explanatory sub-sentences complicate the extraction of important grammatical features, as the explanatory information may also include nouns and adjectives. Thirdly, there are cases where the exquisite linguistic characteristics of the reference language may not fully comply with the SVO rule, e.g. in Greek the subject or the object, or even sometimes the verb of the sentence may be omitted.

### **4.3 Grammatical Analysis- Stemming**

Grammatical analysis tries to exploit grammatical characteristics that may be apparent in a reference language. Grammatical analysis is used to analyze the underlying rules of word construction in a language. The most common application is stemming. Stemming importance has been underlined by (Scott & Mattwin, 1999). Through stemming, the lemma of each word is identified, omitting the part that alters over different word forms.

### **4.4 Grammar or Syntax?**

Both systems use grammatical analysis in identifying the nouns and adjectives that are present in a sentence rather than syntactical analysis. The reasons why grammar is chosen as the basis for this process over syntax include Greek language particularities as well as computational cost efficiency. Greek language (as well as other Indo-European languages), while generally conforming to SVO patterns, enables the

inclusion of supplementary information both in the Subject and in the Object of a sentence. Thus, instead of having a simple noun or an adjective-noun combination serving as a Subject or an Object of sentence, a whole noun phrase may be observed. This noun phrase may rarely be found in exactly the same form over a set of sentences. In addition, Greek grammar states that all the words that compose the noun phrase must follow the appropriate grammatical deviation (exceptions are articles, adverbs and other non-deviated grammatical elements). Therefore, even if a noun phrase is found in more than one sentence, its identification still requires grammatical analysis on each of the words composing it. Taking, for example, the available Greek POS tagger from the Athens University of Economics and Business (AUEB) (Greek POS Tagger), and using it to identify certain sentence word characteristics yielded ambiguous results, regarding the identification of the syntactical sentence features. However, grammatical analysis is heavily dependent on the reference language. Greek language, for example, provides us with numerous rules that may be exploited in extracting nouns and adjectives. Given that, from all the words constituting the Greek language, only articles, adverbs, proverbs and prepositions are considered relatively small in order to be included in a stop list, the main effort is to discriminate between adjectives, nouns and verbs. Greek verbs generally follow an irregular form of grammatical deviation.

<b>Greek Verb</b>	<b>Translation</b>	<b>Simple Past</b>	<b>Stem</b>
Μένω	Stay	Ἔμεινα	με-
Δένω	Tie	Ἔδεσα	δε-
Πλένω	Wash	Ἔπλυνα	πλ-

Table 1. Greek Verbs

As can be seen in Table 1, while stemming in verbs can become very complex, considering irregular verb deviation, or different forms of deviation of seemingly identical verbs, the identification of a verb is a relatively easy task as the verb endings are unique. Moreover, adjectives and nouns follow the same deviation rules, while they share common word endings. Therefore, their identification is equally simple. Thus, in the cases of a noun phrase, noun and adjective stems, rather than the whole

phrase, are considered. A potential drawback of this approach derives from the fact that there are cases where nouns are used as adverbs.

Both of the systems described in later sections are built around a grammatical analyzer. The grammatical analyser is based on work already undertaken in word stemming using grammatical characteristics of Greek language. Initial work on stemming was undertaken by (Porter, 1980), where he gathered the different word endings both for regular and irregular English words and proposed an approach for separating the word ending from the stem itself. Stemming for the Greek language has also been proposed (Kalamboukis, 1995). His work included an adaptation of Porter's approach using Greek word endings.

The utilization of stemming as a lemmatizer and noun and adjective identifier is common between both TCASGL and GUTS. The procedure followed utilizes initially a stop list of words other than nouns (e.g. proverbs, adverbs, prepositions etc), identifies potential non-noun endings (e.g. verb endings) and identifies the unique noun and adjective endings. All the words in the stop-list are eliminated along with the ones whose ending belongs to the verb endings. The resulting word set is ideally composed of nouns and adjectives; however other grammatical elements such as adverbs may also be included.

```

1. Let random word set wD
2. For each l in wD
3.   If l is an Article Then remove l End If
4.   If l is a Preposition Then remove l End If
5.   If l is a Pronoun Then remove l End If
6.   If l is an Adverb Then remove l End If
7.   If l is a Verb Then remove l End If
8.   If l is a Noun Then
9.     If l is a Participle Then
10.      remove l
11.    Else
12.      keep l
13.    End If
14.  End If
15. End For
16. Gather updated table of words wD

```

Figure 4. Stemming and Noun/Adjective Identification

Appendix A provides the list of common Greek language noun, adjective and verb endings, as well as the stop-list used in both systems.

**4.5 Synopsis**

In this Chapter, a presentation of the main linguistic characteristics that impose specific limitations to the development of both TCASGL and GUTS systems is made. Through literature review, it is clarified that in order to extract summaries that it is important to estimate the topic sentences of the documents. Based on that, the parts of speech (nouns and adjectives) considered as important for summarization tasks were identified. Thus, further research was undertaken to decide on the best approach in identifying nouns and adjectives. From the two available approaches, syntactical and grammatical analysis, the latter was selected, due to Greek language particularities.

The next chapter describes the theory behind TCASGL, taking into consideration the conclusions of the grammatical analysis described in this Chapter.

## 5. Text Classification Assisted Summarization for Greek Language - TCASGL

*Chapter 5 describes the TCASGL algorithm and system. It details the steps behind the TCASGL algorithm, provides a description of the system architecture and the modules comprising it, as well as giving important information regarding its classification and summarization tasks. The Chapter concludes with an in-depth analysis of the methodology and algorithms.*

---

### 5.1 Introduction

TCASGL was developed to verify whether Machine Learning approaches could be effectively utilised in generic single-document summarization. However, instead of the almost standard methodology proposed by previous ML approaches (See Chapter 2) that tried to distinguish whether a sentence is a candidate summary sentence, this analysis focused on document summarization by domain identification. The main idea came from the research of (Barzilay & Lee, 2004). The authors discuss that specific domains of documents (earthquake documents in their case), present specific cue phrases, or have cue words that act either terminologically or are most likely found in respective documents (e.g. the Richter scale denoting the magnitude of the earthquake, which is almost always present in respective news articles). This led to the idea as to whether it would be possible to create dictionaries of such words, and if so try to identify them in a random document. The idea was extended by trying to not only consider terminology, but also weight each of these terms with a significance score when they are found in a domain document. This is, obviously, a classification and summarization problem, while in this case, the difference is that it is not important to promote a sentence as a summary, but promote a document to belong to a specific category, and based on the class it belongs to, to extract the most important sentences.

*A priori*, a careful observation of the previous sentence leads to specific questions that must be answered prior to developing the algorithmic model for such an approach.



- Is it possible to construct automatically classes, and according term weights? What effect does the size of the training document set have on each class independently/ as a whole?
- Is it acceptable for a term to belong to more than one class? If yes, what would the class weight of that term be?
- Can the weighted term dictionary of each class be used to extract sentence significance? If yes, how does a sentence term weight affect sentence significance? Does the fact that a term may belong to another category, as well, have an effect on the significance of the sentence in the selected domain?

These questions were decided to answer the proposal of Barzilay and Lee as well as the implications identified through an observation of these remarks. More specifically, the construction of document classes and extraction of potentially weighted terminology addresses the issue of domain knowledge. However, some domains have larger terminology sets than others, so this should also be taken into account. It was also observed during the formulation of the problem that certain terms might be important into more than one domain. Thus, estimating the importance of each term on each of the domains is very important in summary extraction. Yet, the last question addresses the issue of estimating the contribution of a sentence, rather than a term, into a domain, on the premises that a sentence is a set of terms.

The answer to these questions formulated the basis upon which TCASGL was build. Analysis showed that automatic creation was possible with the use of a preclassified training set. Each extracted term (noun or adjective following the analysis presented in Chapter 3) may occur in more than one class, with different significance per class. The term “prime minister”, for example, is more significant in a document considering politics, than in a document considering sports. After deciding on the class a document belongs, the identification of the significance of each sentence is made according to the class dictionary. However, since each sentence is made up of weighted terms in different categories, it is important to use a normalised weight that considers the relative weight for that class, since the same sentence will be weighted differently for a different class.

### 5.2 TCASGL approach

The Naive Bayes assumption of statistical independence states that words in a document are statistically independent with regards to their appearance, and has been proven to work surprisingly well, considering the initial false assumption. However, NBC classifier suffers from bias over unequal in length classes (as proven by (Rish, 2001)). For TCASGL, a single-label, newspaper article-based supervised trainable classifier for the Greek language that uses a normalized approach in assigning a random document to a class was developed. The system takes into account term frequency, along with the size of each class. Therefore, each class component has a normalized weight coefficient, participating in the final classification (and summarization) step. The main difference from common approaches, such as the NBC, is that the classifier does not utilize a product methodology in computing the importance of words in a document to extract the category a document belongs to, but rather the sum of each of the weight coefficient of each word for a category. This approach considers that each word is statistically independent in occurring in a random document, and instead of trying to identify word co-occurrence as in NBC and LMs (computing the product of each probability of word occurrence, therefore searching for word co-occurrence), the contribution of each independent word to the class of the document is estimated. Given a word in a document, the probability of the document belonging to a specific category featuring that word is calculated, and a further estimation on the expected value (mean probability) of the contribution of the entire word set of the document to each one of the available categories is made. A direct outcome from using the sum of probabilities, rather than the product, is that, while the system targets on single-label classification, its decision rule is not as strict as the one defined in NBC, and therefore it can be used for multi-label classification tasks, as well. Another outcome in this approach is that a potential classification error is not propagated in the sum of probabilities as quickly as in the product of probabilities, thus making the algorithm less susceptible to noise.

#### The problem

Let  $i$  be a random noun, and  $D$  a random document featuring word  $i$ , and  $j$  a category the document may belong to. The classifier computes the possibility for document  $D$  to belong to category  $j$  given that word  $i$  appears in  $D$ .

The weight of word  $i$  in category  $j$  ( $w_{i,j}$ ) is computed by (Eq. 23)

$$w_{i,j} = \frac{|tf(i,j)|}{\sum_{i=0}^n |tf(i,j)|} \quad (\text{Eq. 23})$$

where  $tf(i,j)$  the term frequency of word  $i$  in category  $j$ , and  $n$  the total number of unique words comprising category  $j$ .  $w_{i,j}$  in this context denotes the importance or contribution of noun  $i$  in category  $j$ . By dividing by the total number of the noun term frequencies of category  $j$ , a normalized weight factor for nouns in that category is formed, in order to overcome NBC bias. This formula is influenced by the Term Frequency (TF) used in Information Retrieval algorithms (the first parameter of *tf.idf* algorithm). This approach also enables the system to properly estimate the similarity of a random document to a category, since a great number of significant words of a category in a random document (high-weighted word observation), denotes greater similarity between the document and the category. The similarity factor of TCASGL is based on the probability of a random document  $D$  belonging in category  $j$ , if word  $i$  is present in document  $D$ . Given that word  $i$  is assigned a weight per class  $j$ , then this probability is calculated by

$$p_{i,j} = P(D \in j | i \in D) = \frac{w_{i,j}}{\sum_{j=0}^n w_{i,j}} \quad (\text{Eq. 24})$$

where  $n$  is the total number of categories the system can identify. This metric takes into account the cross-class importance of the words. This algorithm strongly resembles the Inverse Document Frequency metric used in Information Retrieval (the second parameter of the *tf.idf* algorithm).

The evaluation criterion that denotes a document belonging to a category is called similarity factor (*sf*) and is calculated by

$$sf(D, j) = \frac{\sum_{i=0}^{Dwords} \frac{w_{i,j}}{\sum_{j=0}^m w_{i,j}}}{Dwords} \quad (\text{Eq. 25})$$

where  $m$  is the total number of categories available and  $Dwords$  the word count of document  $D$ . Since  $sf$  is computed on the observed set of nouns extracted by the document, and not all words appear in category  $j$ , the contribution for each word present in the document is either 0 if the word is not present in category  $j$  or calculated according to (Eq. 24). Dividing by  $Dwords$  gives us the Expected Value for each category  $j$ . The system decides that the document belongs to the category which maximizes the similarity factor  $sf$

$$D \in j \Leftarrow \text{argmax}(sf(D, j)) \quad (\text{Eq. 26})$$

where  $\text{argmax}(sf(D, j))$  is the maximum similarity factor of document  $D$  in categories  $j$ . For example, TCASGL currently identifies six potential categories of articles namely Business and Finance, Politics, Culture, Sports, Technology and Health. A word that is part of all six datasets is the word “πολιτικ” – the stem of word politics or politician. Applying formula (Eq. 23) on word stem “πολιτικ” based on information gathered from the training phase, the weight per category is:

Category	Occurrence	Total Words	Category	Weight
Business and Finance	196	59777		0.149275137
Culture	179	26932		0.302586779
Health	1	3073		0.014815045
Politics	664	63218		0.478181589
Sports	11	25429		0.019693773
Technology	12	15412		0.035447676

Table 2. Weight of word “πολιτικ” in every category

These results indicate that given a document containing only the word “πολιτικ”, then it would be classified in category “Politics” since it holds the greatest similarity factor among every category – the weight of the word being the similarity factor of the document to each category in this case.

Summarization in TCASGL is applied after the classification of the document. The summarization module considers the weight score calculated in (Eq. 24) as the primary sentence selection feature, while extensions using spatial document

characteristics as the ones proposed by (Baxendale, 1958) were also considered. The last feature used in TCASGL is sentence-length. The summarizer uses the class dictionary to assign a proportional term weight as computed in (Eq. 24). In order to extract the sentences that will be kept in the final summary, the summarizer uses the normalized sum of all term weights of a sentence. This is computed by

$$Score_{sent} = \sum_{sent} \frac{\sum p(i,j)}{|p(i,j)|} \quad (\text{Eq. 27})$$

The normalized average term length is used as the bias elimination feature of larger sentences. Thus, only sentences consisting of more important words in a proportional manner are considered. The final length of the summary is manually decided, while the summary consists of the k-th highest scoring sentences sorted by their original document appearance.

The positional feature of TCASGL considers the relative paragraph-sentence position. As it has been experimentally validated (Nenkova, 2005), the most important – summary-wise – sentences in newswire corpora are the initial sentences of the document (the reason that not many systems outperform the basic single document summarizer of DUC-2004). Following that, and (Baxendale, 1958) research, TCASGL uses the relative position of a sentence in a paragraph to compute sentence weight, assigning a greater prejudice score to sentences, located higher in the paragraph. The formula used is depicted in (Eq. 28)

$$PrScore_{sent} = \frac{(a - ((a-1) * k))}{paragraphsize - 1} * Score_{sent} \quad (\text{Eq. 28})$$

where k is the k-th sentence of the paragraph and  $\alpha$  a predefined importance factor.

### 5.3 TCASGL Methodology

Based on the aforementioned analysis (Chapter 5.2) the extracted system is made up of four core modules: the stemmer (Chapter 4), the class training module (executed only once to train the classifier and summarizer), the classifier module, and the summarization module (with or without positional characteristics). This process can

be depicted in Figure 5. Initially the Category creation module preprocesses a set of category documents and calculates the word weights per category to create the category dictionaries. These are used both in the classification and in the summarization modules. A test document is stemmed and its sentences are split. The second step is to calculate the category in which the document belongs to, while the result of this calculation specifies the category dictionary to be used in the summarization.

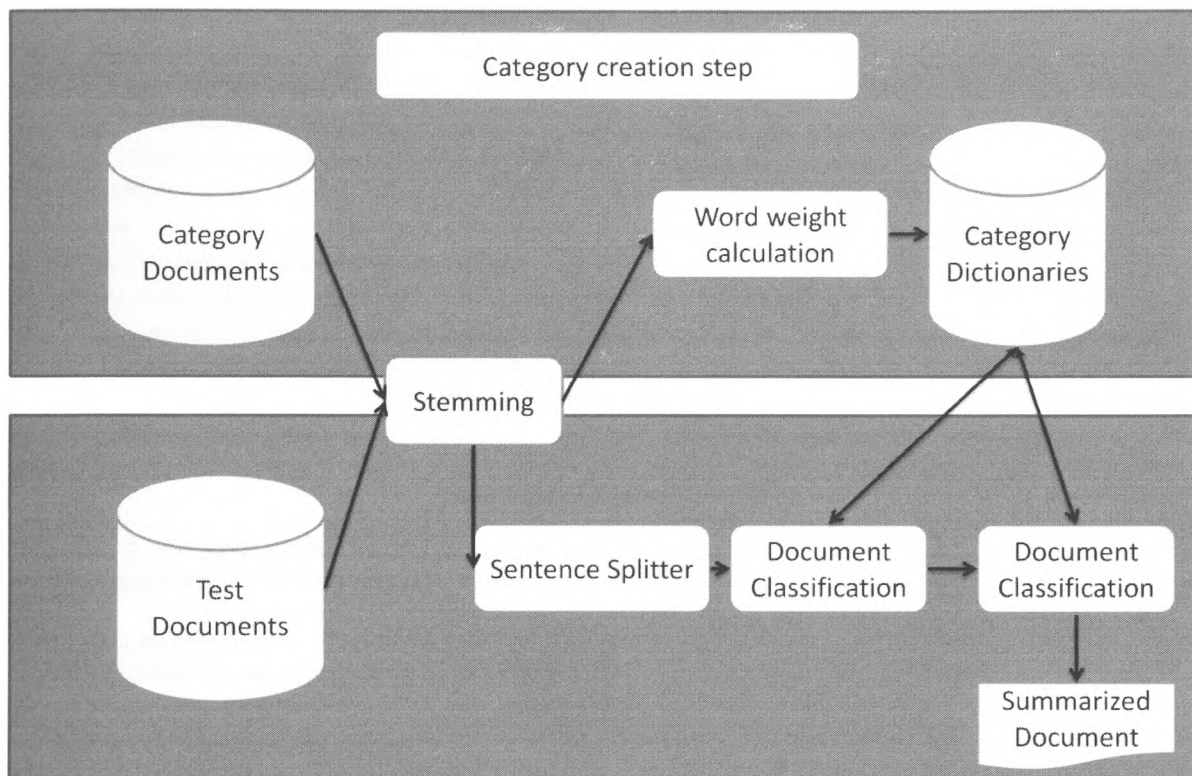


Figure 5. TCASGL Overview

The rest of the subsection presents the algorithms of each module.

### 5.3.1 Training Step

The training step is responsible for building up the dictionaries of words used in classification. Each newspaper article in each category is stemmed and a noun-table per processed article is used to create the weighted category dictionary. Each word is assigned a normalized weight according to the number of nouns present in each category. The training module algorithm is depicted in the following table

1. For each category  $j$  formed by documents  $D_j$
2.   Create an empty dictionary for category  $j$  ( $dict_j$ ), where each dictionary entry is a quadruple (word, cat\_freq, weight, probability) \*
3.   For each document  $D$  of  $D_j$
4.     Stem  $D$  and acquire word table  $wD$
5.     Create a vector  $vD$  of couples (word, doc\_freq) \*\*
6.     For each couple  $c$  of  $vD$
7.       If  $dict_j$  (the dictionary of  $j$  category) contains  $c.word$  Then
8.         Update occurrences (.cat\_freq) of the corresponding couple of  $dict_j$
9.       Else
10.        Append a new quadruple to  $dict_j$  initialized as ( $c.word$ ,  $c.doc\_freq$ , 0, 0)
11.       End If
12.   End For
13. End For
14. Compute  $S_j$  as the sum of all .cat\_freq (frequencies) in  $dict_j$
15. For each entry  $e$  of  $dict_j$
16.   Compute  $e.weight$  as  $e.cat\_freq$  divided by  $S_j$
17. End For
18. End For
19. For each category  $j$  formed by documents  $D_j$
20.   For each entry  $e$  of  $dict_j$
21.     Compute the sum of .weight items of every dictionary entry of every category that has .word item equal to the present word entry ( $e.word$ ) \*\*\*
22.     Compute  $e.probability$  by dividing  $e.weight$  with the sum
23.   End For
24. End For

Figure 6. Training Module Algorithm

\* cat\_freq is the total number (sum) of occurrences of word inside all documents of category  $j$ , weight is computed according to formula (Eq. 23), at step 16, probability is computed according to formula (Eq. 24), at step 22.

\*\* doc\_freq is the frequency of word inside  $D$

\*\*\* step 21 computes the denominator of formula (Eq. 24)

### 5.3.2 Classification Step

The classification step is responsible for assigning a random document to one category. Each random document is stemmed and the resulting noun table is checked against each category to form the similarity factor. The category a document belongs to is defined by the maximum similarity factor. The classification step algorithm operates as follows:

1. Let document to be classified  $D$
2. Stem  $D$  and acquire word table  $wD$
3. For each category  $j$
4.   Initialize the similarity measure of document-category  $sfD_j$  ( $sfD_j \leftarrow 0$ )
5.   For each word  $i$  of  $wD$
6.     Locate the entry  $e$  of  $dict_j$  that corresponds to word  $i$
7.     If the location is successful ( $dict_j$  has an entry for the given word  $i$ ) Then

8. Add to similarity measure of document-category  $sfDj$  the word probability for this category  $pi,j$   
 $(sfDj \leftarrow sfDj + e.probability)$
9. End If
10. End For
11.  $sfDj \leftarrow sfDj \div sizeof(wD)$
12. End For
13. Return  $j$  that has the maximum  $sfDj$

Figure 7. Classification Step

### 5.3.3 Summarization Step

The summarization step acquires word weights per sentence, extracts the topmost important sentences within the limits set, and rearranges the sentences in their original order. The summarization module is presented in the Figure 8.

1. Let document  $D$ ,  $k$  percentage of summaries and class weight lexicon  $l_c$
2. Split  $D$  into array of sentences  $S_D$
3. For each  $s$  in  $S_D$
4. Let sentence weight  $s_w = 0$
5. For each word  $w$  in  $S_D$
6. If  $w$  belongs to  $l_c$
7.  $s_w = s_w + l_{cw}$
8. words = words+1
9. End If
10. End For
11. End For
12. Sort sentences descending according to  $s_w$
13. Keep top  $k$  sentences
14. Sort  $k$  sentences according to appearance in original document

Figure 8. Summarization Module Algorithm

## 5.4 Synopsis

In this Chapter, the presentation of the theory, methodology and algorithms of TCASGL is made. The Chapter focuses on both classification and summarization approaches providing insight on the proposed methodologies both from a theoretical point of view and on the algorithms that resulted in TCASGL system. Specifically, in this Chapter the following subjects were discussed:

- Theoretical validation of the problem of summarization through classification
- The approach proposed for each TCASGL module
- TCASGL training phase (executed once, required for both classification and summarization steps)
- TCASGL classification methodology and algorithms



- TCASGL summarization methodology and algorithms
- Feature set extension for TCASGL summarization considering document characteristics

The next Chapter discusses GUTS system by: formulating the problem of engaging shallow document analysis in document summarization; proposing the appropriate approach introducing novel document features; and presenting the methodology and algorithms of GUTS.

## 6. Generic Unsupervised Text Summarization - GUTS

*Chapter 6 describes the GUTS system, its algorithm, methodology and results. Initially, the theoretical model upon which the GUTS system is based is analyzed, with emphasis on the notion of conceptual flow. An extensive analysis of the methodology is provided, while the chapter concludes with the presentation of the algorithms and an analysis of the process flow.*

---

### 6.1 Introduction

GUTS is based on linguistic characteristics of written documents. It utilizes a new, and exploits already known, sentence selection features. The new feature utilized by GUTS is called “conceptual flow”, while term-co-occurrence, sentence length and relative paragraph-sentence position are also tested – using a similar approach to the one proposed in TCASGL.

GUTS’ approach in single document summarization tries to capture “topic shifts” occurring in the any given document based on the evaluation of the semantic similarities between sentences. Its algorithm does not utilize any external training corpus, relying only on term frequency (how many times a term occurs in the document), term density (how many important terms are existent in a random sentence), term bias (biased term co-occurrence) and conceptual flow (an abstract semantic estimation on topic distribution and regression). These features have been selected as most of them have been extensively used as the basis of many NLP document summarization systems. However, conceptual flow is a newly introduced feature.

Conceptual flow tries to estimate how the authors organize their document. As a feature, the latter tries to identify when a topic starts to be discussed within a document and when the author decides to change topics. A paragraph approach, while

it may describe efficiently this relation between sentences, is based on the assumption that a paragraph is full conceptual entity, while in many cases more than one topics might be considered. In addition to that, conceptual flow tries to estimate topic regression – when topics are discussed again. In order to achieve so, the algorithm considers intra-sentence word occurrence, as well as biased word co-occurrence through Jaccard similarity, combining them linearly.

In the first step of the algorithm an evaluation of whether topic shifts occur in consecutive sentences is made, while later stages perform cross-document semantic similarity calculations. GUTS is inspired by the writing style of authors in presenting information in a document or article, and more specifically the exploitation of the sequential topic shifts that occur between segments of analysed concepts. These topic shifts are closely related to the presentation and validation of the author's ideas or remarks. Thus, GUTS' goal is to capture these changes in the discussion of topics and evaluate the importance of each concept. Conceptual flow, as a feature, following a linguistic analysis, refers to the construction of the set of consecutive sentences that describe a common context. In a similar manner to what has been proposed by (Zha, 2002) regarding sentence clustering and sentence importance through common words and consecutive sentences – the major difference is that GUTS performs a semantic analysis on words rather than applying mutual reinforcement principle - the system tries to estimate how word co-occurrence in sentences denotes semantic relation and how this affects consecutive sentences. In addition to that, the algorithm has been developed keeping in mind that in a particular text, more than one topic may be discussed. Therefore, it is vital not only to identify the primary topic, but also identify secondary topics and evaluate their importance in the document. The algorithm performs on a progressive approach. Initial (primary) topic clustering occurs on consecutive sentences, and the most important sentences are extracted to form an initial summary. Topic clustering in this manner is used abstractly. The reason for that lies in the fact that a topic may not be thoroughly discussed within a primary topic cluster, but within a series of non-consecutive topic clusters. Full topic clustering occurs during the second step of the system, where the initial summary is scanned for

topic repetition. If the system identifies the full topic cluster map of the document, it extracts in a common manner the final summary, while failure to identify additional topic clusters results in a statistical approach based on term density. Therefore, the random document is scanned at most three times in order to identify the full topic cluster map of the document. Failure to achieve the expected size of summary, results in a simple statistical approach based on normalized term frequency, where each sentence is evaluated through term density.

## 6.2 GUTS Approach

In order to identify the topic clusters of the document, GUTS follows a three-step approach.

### 6.2.1 Semantic lexicon creation

Initially, a term analysis that captures term frequency and term co-occurrence is performed. Term frequency is important in identifying words that may be document terminology. Term co-occurrence, on the other hand, is used as the semantic similarity measure that identifies the bias between word occurrences. Both measures are used in evaluating sentence similarity. An  $n \times n$  matrix is created,  $n$  denoting the number of individual words present in the document. The words are sorted by term frequency. Each  $(i, j)$ -th entry in the matrix holds the semantic weight between  $\text{term}_i$  and  $\text{term}_j$ . The elements in the diagonal of the matrix hold the term frequency. The semantic similarity weight is based on the Expected and Observed Co-occurrence Probabilities.

- Observed Co-Occurrence ( $OC_{ij}$ ):  $OC_{ij}$  denotes the observed common occurrences of words  $i$  and  $j$  ( $i \neq j$ ) in a document with a fixed number of sentences  $s$  and it is calculated by

$$OC_{i,j} = \text{occu}_{i,j} / s \quad (\text{Eq. 29})$$

where  $\text{occu}_{i,j}$  is the number of common occurrences between words  $i$  and  $j$ .

- Expected Co-Occurrence ( $EC_{ij}$ ):  $EC_{ij}$  denotes the expected common occurrences of words  $i$  and  $j$ , considering statistical independence in the existence of

words  $i$  and  $j$ . Thus, using the Bayesian Rule of Statistical Independence the Expected Co-Occurrence is calculated by

$$EC_{i,j} = \frac{\frac{occur_i * occur_j}{s}}{\frac{occur_i}{s} + \frac{occur_j}{s}} = \frac{occur_i * occur_j}{s * (occur_i + occur_j)} \quad (\text{Eq. 30})$$

If the  $OC_{i,j}$  is greater than  $EC_{i,j}$  the system considers that there is a bias in the occurrence between these words and therefore, that words  $i$  and  $j$  are semantically related.

Thus, the semantic similarity matrix elements are:

$$SR_{i,j} = \begin{cases} 0, \text{ if } OC_{i,j} < EC_{i,j} \text{ and } i \neq j \\ OC_{i,j} - EC_{i,j}, \text{ if } OC_{i,j} > EC_{i,j} \text{ and } i \neq j \\ \text{Occurrence of word } i, \text{ if } i = j \end{cases} \quad (\text{Eq. 31})$$

### 6.2.2 Conceptual Flow Topic Creation and Summarization

The clusters are created according to the similarity scores of consecutive sentences. The Sentence Similarity score ( $SS_{si,si+1}$ ) of sentences  $si$  and  $si+1$  is computed using the SR of the words of both sentences. The formula used to calculate cross-sentence relationship is based on a modification on Jaccard similarity and is calculated as

$$SS_{si,si+1} = \sum_{i=1}^m \sum_{j=1}^n \frac{SR_{i,j}}{m+n} \quad (\text{Eq. 32})$$

where  $m$  is the words of sentence  $si$  and  $n$  the words of sentence  $si+1$ . This score denotes the semantic similarity of two consecutive sentences as it does not only identify common word occurrence but also weighted semantic relationship of random words. In order to extract the document clusters, a threshold feature should be used on the  $SS$  scores of each consecutive sentence. This threshold states the limit below which two consecutive sentences are considered semantically irrelevant. In order to calculate the Similarity Threshold ( $ST$ ), the system uses the local minima of cross-

sentence similarities (if projected on a graph) and calculates the average of these minima (Eq. 33).

$$ST = \frac{1}{i} \sum_i (SSmin_{si,si+1}) \quad (\text{Eq. 33})$$

where  $SSmin_{si,si+1}$  denotes the semantic similarity score of sentences where a local minimum is observed.

The basis of the system upon which sentence similarity is considered states that all sentence links below  $ST$  denote a topic shift between the sentences, and therefore define the edge of each of the topic clusters of the document. Each topic cluster is made up of consecutive sentences that have a Sentence Similarity greater than  $ST$ . The extracted summary is made up by sentences sorted by appearance in the initial document from all the topic clusters identified in the document. If the topic cluster is composed of a single sentence, this sentence is included in the final extract, while remaining summary sentences - the sentences that remain after single sentence topic clusters have been considered - are picked by remaining topic clusters on a proportional rate. The score  $Sk$  of the  $k$ -th sentence is calculated using the number of occurrences of the non-unique document words (words that appear more than once) in sentence  $k$ .

$$Sk = \sum_{i_n} occur(word_{i_n}) / len(k) \quad (\text{Eq. 34})$$

where  $i_n$  is the  $n$ -th word of sentence  $k$ ,  $occur(word_i)$  is the number of occurrences of the  $i$ -th word and  $len(k)$  the number of terms in the  $k$ -th sentence.

If the number of extracted sentences is greater than the percentage of summarization achieved, the topic cluster procedure and summarization is repeated on the summary. However, instead of forming a new abstract semantic lexicon for the summary, the algorithms utilize the one computed in the initial step of the algorithm. This last step is repeated until the algorithm fails to further shorten the extract, maintaining both the original similarity matrix and  $ST$  in every step.

The aforementioned methodology captures the conceptual flow sentence selection feature. While conceptual flow is not used for sentence selection, it clusters, at first, consecutive and, secondly, non consecutive sentences into semantic sets. These sets have direct and inferred word similarity through the evaluation of common word occurrence and biased word probability across sentences. Thus, this way not only the flow of concepts is captured, but also after a first filtering, the remaining concepts are estimated for topic regression.

### 6.2.3 Intra-document Summarization

In case that the number of extracted sentences is greater than the required threshold set by the user, and the initial algorithm has reached its limits in extracting the appropriate sentences, the same approach is engaged on an intra-document level. Instead of forming topic clusters on consecutive sentences, the topic cluster algorithm is applied on non-consecutive sentences and thus tries to identify the unique topic clusters of the whole document ( $SS_{si, si+k}$ , where  $k>1$ ). The Similarity Threshold ( $ST$ ) is maintained from the initial calculations, along with the similarity matrix. The reason this algorithm is applied on the first step summary rather than the initial document is the calculation cost of this approach, as it requires an exhaustive comparison between each sentence of the document. In this case a sentence may be part of more than one cluster. However, its significance will be considered per topic cluster independently, so as to capture and evaluate potential secondary topics discussed in each sentence. In cases where the algorithm fails to achieve the required summary length, the system falls back to a statistic based approach based on term density. Taking into consideration the original word occurrence, the algorithm computes the sentence weight as the average term frequency of each sentence (Eq. 35) regardless of the non-existence of clusters. The final extract is composed of the highest scoring sentences sorted by sentence appearance in the original document.

## 6.3 Methodology

Following the establishment of the formulas and the standardization of GUTS' approach on topic clustering, the system was built using six distinctive modules. These modules are:

- Stop-list Creation and Grammatical Analysis (stemming) module
- Abstract Semantic Relation Matrix module
- Conceptual Flow Topic Cluster identification module
- Intra-document Topic Cluster identification module
- Topic Cluster Summarization module
- Term Density Summarization module

The following Figure shows an outline of the full methodology.

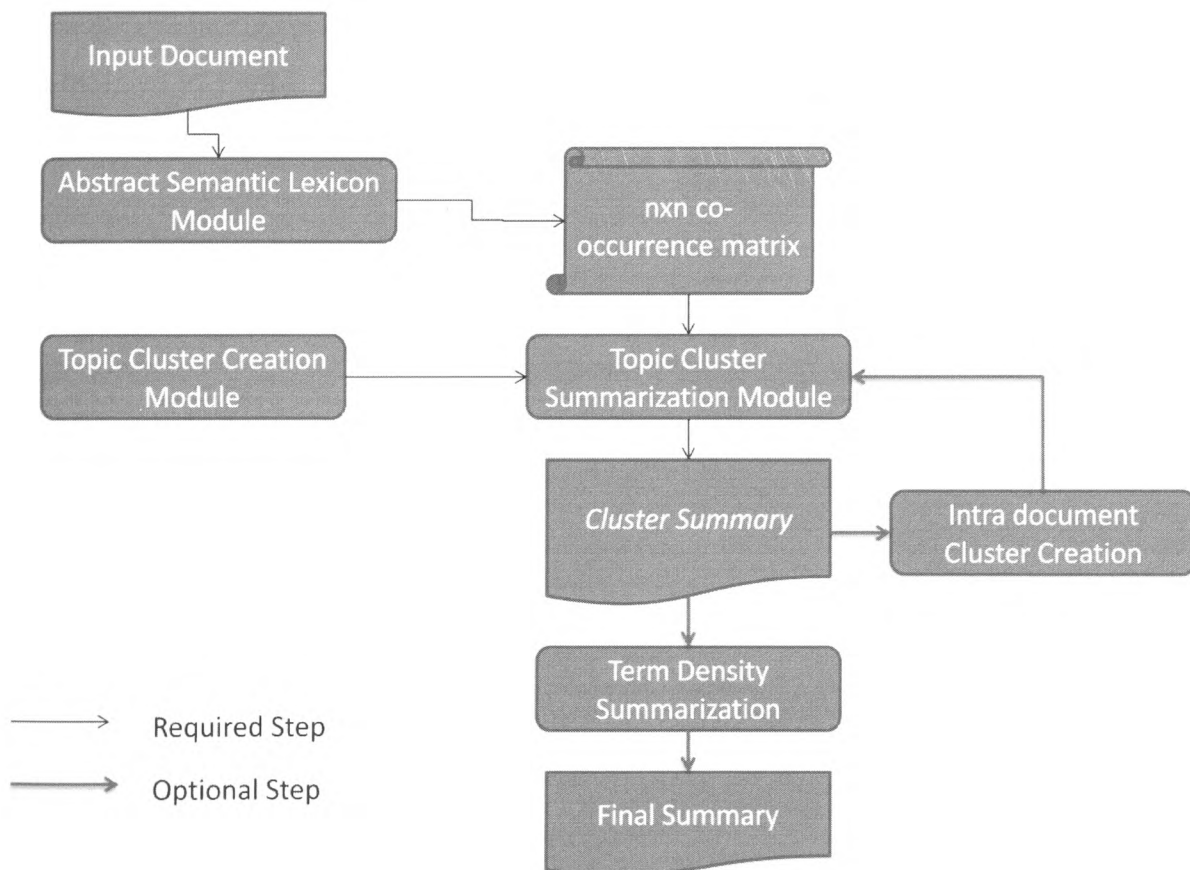


Figure 9. GUTS Overview

Stop-list Creation and Grammatical Analysis and Abstract Semantic Relation Matrix modules are only used once in the system in the initial phase of the algorithm, while the remaining modules are used as many times as required, according to the proposed approach.



Following the Grammatical Analysis module (Chapter 4), the Abstract Semantic Relation matrix module is used to create the relation matrix between important words that are found in common sentences. Initially, each of the stems extracted from the Stop-list creation and grammatical analysis modules are checked on a sentence-by-sentence level and word occurrences are counted, as well as the sentences in which each one was found. This forms the diagonal of the Abstract Semantic Relation matrix. Each word in the matrix is then checked against any other word and the Expected and Observed Probabilities are calculated per word pair. If the significance condition for the semantic relation between words is satisfied (Observed Co-occurrence > Expected Co-occurrence), then the system constructs an Abstract Semantic Relation between the words calculated by (Eq. 30). The algorithm is summarized in Figure 10.

```

1. Let document to be summarized D
2. Split D into sentences, forming sD
3. Create ssmatrix(termi, termj, weight)
4. For each s in sD, split s into words, forming wD
5.   For each w in wD
6.     Stem w and create stem table stD(stem, occurrence, sDindex)
7.   End For
8. End For
9. For each stemi in stD
10.   For each stemj in stD
11.     If stemi.equals(stemj)
12.       Update stDi.occurrence
13.     End If
14.   End For
15. End For
16. Update ssmatrix(stemi, stemj, stDi.occurrence)
17. For each couple stems c in table stD
18.   Create co-occurrence index cind
19.   If c.sDindex1 = c.sDindex2
20.     Increment cind
21.   End if
22.   Calculate ECc and OCc
23.   If ECc ≥ OCc or OCc = 0
24.     Update ssmatrix(stemi, stemj, 0)
25.   Else
26.     Update ssmatrix(stemi, stemj, OCc - ECc)
27.   End If
28. End For

```

Figure 10. Abstract Semantic Relation Matrix Creation

The resulting Abstract Semantic Relation matrix is not reconstructed during the Conceptual Flow/Intra-document Topic Cluster Identification or Topic Cluster/ Term Density Summarization modules. The initial matrix is used instead.

The next step is to apply the Conceptual Flow Topic Identification algorithm (Figure 11) between consecutive sentences, as this will give an initial reduction on sentences, prior to applying this technique on a cross-document level. For each pair of consecutive sentences the Sentence Similarity is computed, based on the weighted connections of stems from the Abstract Semantic Relation matrix. After all Sentence Similarity metrics have been computed, the system identifies the local minima of the resulting schematic and computes the average of these minima to acquire the Similarity Threshold. Thus, a set of Topic Clusters is extracted. The Topic Cluster (Figure 12) summarization module is called and an initial summary is produced. If the summary is larger than the desired, the Intra-document Topic Cluster module is used. If the Intra-Document Topic Cluster module (Figure 13) fails to produce the expected length of summary, or does not find any further topic clusters, the Term-Density Summarization module (Figure 14) is utilized to ensure both that the system will not perform infinite loops in extracting the summary and that the expected size of the summary is reached.

```

1. Let document D to be summarized, and corresponding semantic similarity matrix ssmatrix(stemmn,
stemn, weight)
2. Split D into sentences, forming table sD
3. Create empty sentences vector V(sentence,sentence_terms)
4. For each sentence s in sD
5.   Split s in words, forming wD
6.   For each word w in wD
7.     Stem w and append in SV.sentence_terms
8.   End For
9. End for
10. Create Similarity Score vector SS(si,sj,Sscore)
11. For each couple of consecutive sentences c in SV
12.   For each stem i in c.stems1
13.     For each stem j in c.stems2
14.       Sscore = Sscore + ssmatrix.weight
15.     End For
16.   End For
17. End For
18. If Similarity Threshold ST does not exist
19.   Create ST
20.   Create similarity score vector SSmin
21.   For each similarity score ss in SS
22.     If ss is local minimum
23.       Append Sscore in SSmin
24.     End If
25.   End For
26.   ST = average(SSmin)
27. End If
28. Create Topic Cluster vector TC(sentences)

```

```

29. For each sentence pair c in SV
30.   If TC is empty
31.     Add c.sentence1 in TC
32.   End If
33.   If SS.Score>ST
34.     Add c.sentence2 in TC
35.   Else
36.     Add new topic cluster entry in TC
37.   End If
38. End For

```

Figure 11. Conceptual Flow Topic Cluster Identification

The Topic Cluster Summarization module is called every time a new Topic Cluster Identification module is called, and reduces the number of sentences in the final extract. Based on the Topic Clusters, it tries to eliminate sentences based on their importance as calculated by word occurrence, while simultaneously maintaining the topics of the document without any loss of information. This implies that a Topic Cluster composed of one sentence, will be carried over to the next iteration, until either eliminated by the Intra-document Topic Cluster Identification module, statistically eliminated by the Term-Density Summarization module, or maintained in the final extract. Thus, in order to provide information on the importance of each Topic Cluster, based on the assumption that larger Topic Clusters are more important than smaller Topic Clusters, the Single Sentence Topic Clusters are initially extracted, while for the remaining clusters, proportionality is followed in extracting sentences, according to the length of the required extract.

```

1. Let document D, and corresponding semantic similarity matrix ssmatrix(stemm, stemn, weight) and
   topic clusters TC
2. Let summary vec sum_vec (sentence index, sentence)
3. Compute TCnot unique = TC where TC.length > 1
4. For each topic cluster TC
5.   If TC.length > 1
6.     For each sentence in TC
7.       Compute Sentence Score according to (Eq. 32)
8.       If length(TC) < length(expected summary)
9.         Append proportionally the most important sentences to sum_vec according to
           (length(summary)-length(TC))*(TC.length/TCnotunique.length)
10.      Else
11.        Append most important sentence and sentence index to sum_vec
12.      End If
13.    End For
14.  Else
15.    Append sentence and sentence index to sum_vec
16.  End If
17. End For
18. Sort sum_vec according to index
19. Let summary s
20. For each sentence in sum_vec

```

```

21.     s+=sum_vec.sentence
22. End For
23. Return s

```

Figure 12. Topic Cluster Summarization Module

If the Topic Cluster Summarization module fails to reduce the length of the extract, compared to the original input text, or does not achieve the expected length of summary, the Intra-document Topic Cluster Identification module (Figure 13) is initialized.

```

1. Let summary Sum, and corresponding semantic similarity matrix ssmatrix(stemm, stemn, weight) and
   similarity threshold ST
2. Split Sum into sentences, forming table sSum
3. Create empty sentences vector V(sentence,sentence_terms)
4. For each sentence s in sSum
5.     Split s into words, forming wSum
6.     For each word w in wSum
7.         Stem w and append in SV.sentence_terms
8.     End For
9. End for
10. Create Similarity Score vector SS(si,sj,Sscore)
11. For each couple of non-consecutive sentences c in SV
12.     For each stem i in c.terms1
13.         For each stem j in c.terms2
14.             Sscore = Sscore + ssmatrix.weight
15.         End For
16.     End For
17. End For
18. Create Topic Cluster vector TC(sentences)
19. For each sentence pair c in SV
20.     If TC is empty
21.         Add c.sentence1 in TC
22.     End If
23.     If SS.Score>ST
24.         Add c.sentence2 in TC
25.     Else
26.         Add new topic cluster entry in TC
27.     End If
28. End For

```

Figure 13. Intra-Document Topic Cluster Identification module

The basic idea behind the Intra-Document Topic Cluster Identification module is the same as the Conceptual Flow Topic Cluster Identification module, but instead of applying the algorithm on consecutive clusters, it is applied on the resulting text, after initial reduction by the previous module. The algorithm of the Intra-document Topic Cluster identification is applied using the initial Abstract Semantic Relation matrix and the same Similarity Threshold as the initial document.

The last, optional, step in GUTS system ensures that when the algorithm fails to further reduce the extracted summary to the desired length, it can still be achievable through the use of statistics. While Topic Cluster Summarization module tries to maintain the original topics and their importance in the document, this step ignores the document topic as they are described in the clusters and considers each sentence as independent to one another. Therefore, the algorithm can migrate from topic importance to sentence importance, where in this case it is computed using term density. Term density refers to the average term frequency of each sentence. In order to compute term frequency, the initial abstract semantic matrix is used. In order to avoid including larger sentences (depicting greater Sentence Scores), as opposed to smaller and probably more important sentences, the average term frequency is also considered. The algorithm is depicted in Figure 14.

1. Let summary Sum, and corresponding semantic similarity matrix ssmatrix(stemm, stemn, weight) and
2. Split S into sentences, forming table sSum
3. Create empty sentences vector V(index,sentence,Sk)
4. For each sentence s in sSum
5.     Compute V.Sk according to (Eq. 34)
6. End For
7. Sort descending V according to Sk
8. Keep the topmost sentences equal to the expected summary length
9. Sort ascending V according to index
10. Let summary s
11. For each sentence in sum\_vec
12.     s+=sum\_vec.sentence
13. End For
14. Return s

Figure 14. Term Density Summarization module

## 6.4 Synopsis

In this Chapter, the GUTS system is thoroughly discussed. A presentation of the features of GUTS system is made, and an extensive description of GUTS system in terms of feature estimation and discrete module identification provided. Each module is described and validated, both independently and as part of the GUTS system. The chapter concludes with the presentation of GUTS' algorithms and extensions that were considered, regarding position characteristics.

## 7. Evaluation – Approaches and Results

*Chapter 7 provides a full account of the available evaluation approaches and their underlying theory. It offers insight on what intrinsic and what extrinsic evaluation is, as well as presenting a series of evaluation conferences that have initiated research in the area of Summarization Evaluation. Moreover, in this chapter a presentation of the efficiency of TCASGL and GUTS is provided both in terms of extrinsic and intrinsic measures.*

---

### 7.1 Document Summarization Evaluation

An important task in summarization is the evaluation of the resulting summaries. Evaluation is a standard procedure that can be intrinsic or extrinsic, as well as manual or automatic. In this section, discussion on both intrinsic and extrinsic evaluation approaches that have been proposed and used extensively is made.

### 7.2 Intrinsic Evaluation

Intrinsic evaluation in generic single-document summarization is a task of quantitative comparison between manual and automatic summarization tasks. Intrinsic evaluation can be either semi-supervised or supervised. Supervised approaches precondition human interference in forming their evaluation scheme, while automatic evaluation metrics do not require human intervention. In order to perform intrinsic evaluation tasks on a summarizer a reference summary set is required. The most prominent and widely used solution is human generated summaries, almost always referred to as gold standards. Gold standards are human summaries of texts of the test corpus, and are generally considered to be the best candidate summaries for a given document. In order to reduce summarizer subjectivity, it is usual to assign manual summarization of documents a number of human summarizers, and acquire the best candidate summary out of the common consensus. Gold standards are used as the comparison

measurement, by which a number of evaluation metrics may be used to test the efficiency of algorithms. Evaluation metrics try to identify the similarity between machine-generated and human summaries, while in later approaches focus is also given to capturing and modelling the human comprehension of summaries.

### 7.2.1 Precision, Recall, F-measure

Precision, Recall and F-measure are three statistical evaluation features widely used in estimating NLP and ML approaches. They are defined as following. Let  $h$  be a human summary and  $m$  be a machine summary. Precision  $P$  is calculated as

$$P = \frac{h \cap m}{m} \quad (\text{Eq. 36})$$

while Recall  $R$  is calculated as

$$R = \frac{h \cap m}{h} \quad (\text{Eq. 37})$$

Precision tries to estimate how many sentences have been identified correctly, disregarding the fact that not all sentences may have been retrieved, while recall tries to estimate how many sentences have been collected correctly, disregarding surplus sentences that may have not been included in the human summary. As may be obvious, these metrics act complementary and therefore it is more important to consider their combination rather than each one of them independently. A combination of these metrics can be approximated using the F-measure. F-measure is generally calculated as

$$F_{\beta} = (1 + \beta^2) * \frac{P * R}{(1 + \beta^2) * P + R} \quad (\text{Eq. 38})$$

where  $\beta$  is the significance parameter on Precision  $P$ . For various values of  $\beta$ , Precision acquires a different value than Recall, while for  $\beta=1$ , Precision and Recall are taken into consideration equally.

### 7.2.2 ROUGE-N

ROUGE-N (Recall Oriented Understudy for Gisting Evaluation) (Lin & Hovy, 2003) is a recall oriented method used to evaluate the efficiency of automatic summarization systems. The method treats words as n-grams and tries to find the maximum number of common n-grams between a candidate summary and a set of references summaries.

$$ROUGE-N = \frac{\sum_{S \in RefSum} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in RefSum} \sum_{gram_n} Count(gram_n)} \quad (\text{Eq. 39})$$

Where  $n$  stands for the length of the n-gram,  $gram_n$  and  $Count_{match}(gram_n)$  is the maximum number of n-grams co-occurring in a candidate summary and a set of references summaries. *ROUGE-N* is not a uniquely identified evaluation metric as the n-gram size applied is used to identify the *ROUGE-N* approach used; *ROUGE-1* uses unigrams, *ROUGE-2* uses bigrams etc. In addition to that the author proposed a series of extensions using alternative evaluation features such as:

- *ROUGE-L*: using Longest Common Subsequence; that is the maximum number of common ordered words found in both extracts, either on a sentence or a summary level.
- *ROUGE-W*: using Weighted Longest Common Subsequence taking into account spatial characteristics in the ordered word occurrence between two summaries.
- *ROUGE-S*: using Skip-bigrams (and pair of words in their sentence order, within a sentence word distance) in evaluation.

*ROUGE-1* in particular is considered to adapt better to the human evaluation of the document as stated by Lin and Hovy (2003).

### 7.2.3 Pyramid

Pyramid Evaluation approach (Harnly, Nenkova, Passonneau, & Rambow, 2005) is a semi-supervised evaluation approach used in the later DUC conferences. The main idea behind Pyramid is significant word spans, which the authors refer to as Summary Content Units (SCU).



A set of ideal summaries is annotated for SCUs, which are gathered and weighted, forming the pyramid. The input document is also annotated with candidate SCUs named contributors. Each contributor is tested against every SCU of the pyramid and a score is extracted, against which the summary is evaluated. The approach utilizes four steps: enumerating the contributors, match the contributors with the most similar SCU of the pyramid, Select the highest scoring contributors, and calculate summary score against SCU scores.

#### **7.2.4 Basic Elements**

A relatively new idea sharing common characteristics with the pyramid (Harnly, Nenkova, Passonneau, & Rambow, 2005) approach, has been proposed by (Hovy, Lin, Zhou, & Fukumoto, 2006) and used in DUC 2005 conference along with ROUGE. According to the authors, a Basic Element (BE) is a small fragment of text such as a noun or verb, or a triplet denoting the relation between a head BE and a dependent on the BE in the form of head/modifier/relation. BE is composed of three modules:

- the BE breaker that identifies BE in text
- the BE matcher that calculates the similarity between two BEs and
- the BE scorers that assign a score to every BE.

The algorithm is composed of two steps:

- a preparation step that considers a set of ideal summaries out of which extracts semantically related BEs and assigns scores per BE and
- a scoring phase where the algorithm performs the same task on the candidate summary using the training BEs for scoring.

The summary is compared with ideal summaries against: lexical identity, lemma identity, synonym identity through Wordnet and semantic generalization.

### **7.3 Extrinsic Evaluation**

In contrast to intrinsic evaluation, extrinsic evaluation explores qualitative characteristics of the summaries. Since qualitative characteristics can not always be

modelled or quantified, intrinsic evaluation is either supervised or operates in conjunction with other tasks. Extrinsic evaluation is applied in areas such as task-oriented summarization, where a summary is produced as a precondition for another task, or query-based summarization where the summarizer is asked to extract a summary that revolves around a specific user question. In the following sections, a presentation of indicative extrinsic evaluation metrics is made with examples on how they have been measured in the past.

### **7.3.1 Usefulness and Responsiveness**

Two of the first extrinsic evaluation features to be used in DUC conferences were Usefulness and Responsiveness (Over, Dang, & Harman, 2007). Both of these metrics required human evaluation. Usefulness was applied on the task of title extraction, and the human evaluators using a scale of 1 through 5, were asked to score each extracted title according to whether it will help them pick that specific document or not. Responsiveness, on the other hand, was used to estimate whether the extracted summary satisfies a specific need. Again, the evaluators used a scale from 1 through 5. A similar approach has been utilized by (Jing, Barzilay, Mckeown, & Elhadad, 1998). The authors performed both extrinsic and intrinsic evaluation on a set of candidate summaries that targeted on generic single-document summarization. However, on the extrinsic part of the evaluation the authors proposed a task-oriented evaluation procedure, where the human evaluators graded the efficiency of the machine summary with regards to how well it answers a question regarding the main topic of the document. As they stated, this would be more appropriate if the summarization was query-based, however, they minimized the differences between generic vs. query-based summarization, by asking a generic question on the main topic of the document.

### **7.3.2 Relevance-Prediction**

Relevance-Prediction extrinsic evaluation approach has been proposed by (Dorr, Monz, President, Schwartz, & Zajic, 2005). Relevance-Prediction utilizes human summarization by the evaluator, and was applied to title extraction. Relevance-Prediction tries to capture whether there is agreement between the human extracted

title and some task-oriented or machine generated summary. The basis of the metric lies in two questions: whether they can make consistent judgement between a summary and the full-text document; and whether they can make quicker judgements from the summary rather than the original document. In order to measure Relevance-Prediction, the authors ask the human evaluators to make a judgement on both the document and the summary. If the judgement is the same in both the document and the summary, then it is assigned a score of 1, else it is assigned a score of 0. Let  $s$  a summary,  $d$  a document,  $j(s,d)$  the judgement score on  $s$  and  $d$ , and  $DS_i$  the document/summary pair for an event  $i$ . The formal calculation of Relevance-Prediction is (Eq. 40)

$$\text{Relevance-Prediction}(i) = \frac{\sum_{s \in DS_i} j(s,d)}{|DS_i|} \quad (\text{Eq. 40})$$

### 7.3.3 Complex Approaches

A more complex approach used in proceedings of meetings proposed by (Liu & Liu, 2008), defined a series of 9 individual tasks upon which the human evaluators were asked to give their opinion on a scale of 1 to 5. These tasks dealt with how well the summary reflected the discussion flow, on the coverage of the topics discussed, sentence to meeting relevance, information redundancy, topic accordance ration, role accordance ratio, sentence redundancy, sentence importance, overall estimation of the summary. These features are organized into four groups: Informative Structure, Informative Coverage, Informative Relevance and Informative Redundancy. Each group was considered as an individual feature.

### 7.4 Document Summarization Conferences and Workshops

In addition to evaluation metrics, another driving force in document summarization and system evaluation is task-driven conferences such as Message Understanding Conference (MUC), Document Understanding Conference (DUC), Text Analysis Conference (TAC) that has replaced DUC since 2008, SUMMAC and NTCIR. The importance of these conferences is underlined by the fact that a lot of the research in the area has been presented in these conferences, while evaluation metrics such as the

ones described before have been proposed and used extensively in these conferences. In this section, a brief review of the conferences and their tasks, important results that defined the state of the art as well as the trends in document summarization are presented.

MUC (Grishman & Sundheim, 1996) conference was a series of 7 conferences organized by NRAD, RDT and E division of the Naval Command, Control and Ocean Surveillance Centre, and DARPA. MUC initially targeted at analysing automatically military messages. Despite being named “conferences” MUC were, in fact, a series of evaluations with the participants analysing the efficiency of their systems. MUC is based on task-driven analysis using both training and test data. MUC-1, organized in 1987, did not use any formal evaluation approach. In MUC-2, a template-driven approach was decided, while for each message a series of slots had to be filled in. The evaluation metric used was Precision and Recall. The messages related to naval reports and sightings. In MUC-3 (1991), the messages changed to terrorist reports and the template became slightly more complicated. MUC-4 introduced more complex templates maintaining the same task, while MUC-5 (1993) which was part of the TIPSTER (Mani et al. 1999) program by DARPA, introduced 11 templates with 47 slots to be filled in. In MUC-6 (1995) three goals were set: short-term subtasks, with the aim of developing automatic domain independent approaches, portability and deep understanding measures as co-reference, word sense disambiguation using Wordnet (Miller, 1995) and grammatical analysis. MUC-7 (1995), which was the last MUC conference, organized in 1997, introduced multilingual evaluation, while primary tasks included mainly deep understanding measures such as: Named Entity - proper name disambiguation, Multilingual Entity, Template Element - extracting generic information, Template Relation - information based on specific roles, Scenario Template – extraction of specific information and relation to specific roles and Co reference.

Another series of evaluation conferences sponsored by DARPA were the SUMMAC and DUC/TAC conferences (Over, Dang, & Harman, 2007). The SUMMAC evaluation conference took place in 1998 as part of the DARPA TIPSTER program

and 16 systems participated on two summarization tasks. This provided the initiative for DARPA to sponsor DUC conferences that run from 2001-2007 annually. In DUC conferences a number of evaluation approaches were considered, that have led the state of the art in document summarization during the last decade. In DUC conferences tasks as intrinsic over extrinsic evaluation, single and multidocument summarization (single document has been dropped out of the DUC conferences for reasons explained in (Nenkova, 2005)), summary extraction and abstraction and generic and query-based summarization tasks have been considered. The main contributions of DUC, apart from the different systems and technologies that were presented, were the automatic evaluation procedures, most of which have been used extensively since. For example, all of the intrinsic methods described in the previous section, except Precision and Recall, as well as some of the extrinsic methods, were considered in the various DUC conferences. The last DUC conference took place in 2007, while from 2008 and onwards it became part of the Text Analysis Conference. TAC has been organized by the National Institute of Science Technology annually since 2008 and this year features three tracks: Knowledge Base Population Track, Recognizing Textual Entailment Track and the Summarization Track whose ancestor is DUC. This year's tasks include Guided Summarization Task, Automatic Evaluation, and a Multilingual Task.

The first DUC organized in 2001 offered two tasks: a Generic Single Document Summarization Track with a threshold of 100 words; and a multi-document track with summaries length of 50-400 words. Evaluation was manual.

In the second DUC conference, the multi-document tasks were altered to 10-200 words, while an extract module of 100-200 words was introduced. The third DUC conference in 2003, dropped generic multidocument summarization, and concentrated on headline extraction in generic single-document summarization. In multi-document summarization, focused tasks were proposed as Viewpoint summarization, event-based summarization and topic-based summarization. In the third year of DUC conference, the Responsiveness and Usefulness were introduced as manual evaluation metrics. In DUC 2004, the headline extraction task was maintained as the single

document summarization task, while in multi-document summarization tasks consisted of topic-based and query based summarization tasks. The query-based summarization systems had to answer a simple question of WHO-IS. In 2004, the ROUGE automatic evaluation metric was introduced. Later, DUC conferences completely dropped generic single document summarization tasks, and focused on complex question multidocument summarization. The Pyramids and Basic Elements were also introduced in the evaluation. In the last DUC conference in 2007 the multidocument task, was kept the same, while an update task was introduced, based on multidocument newswire summarization, given that the user has already read a series of documents on a topic, providing the user with updates on the matter.

A similar workshop (NTCIR) is also organized in Japan annually. Its main tasks revolve around general Information Extraction tasks, while Summarization tasks were initiated since the second NTCIR workshop in 2001. Firstly, the summarization tasks considered used manually summarized newspaper articles, considering both sentence extraction and abstraction, while the side task was the proposal of the summary evaluation approaches to be followed for later conferences. Secondly, in NTCIR-3 single and multi-document summarization were considered, while the tasks consisted of generic and query-based summarization. Thirdly, in NTCIR-4, the single document summarization was dropped, while query-based multidocument summarization was considered. Evaluation was both intrinsic and extrinsic and the evaluation metrics used, were Precision, Recall and F-measure. Finally, NTCIR-5 and 6 were the last to feature summarization tasks on multi-document summary extraction.

### **7.5 Evaluation Tools for Current Research**

In this section, a presentation of the evaluation approach for each of the algorithms is presented.

#### **7.5.1 TCASGL Evaluation approach**

TCASGL classification module was tested against two statistical algorithms:

- NBC as provided by Mallet toolkit (McCallum)

- statistical Language Models as provided by Lingpipe (Alias-I) natural language processing toolkit.

The statistical Language Models utilized included a 6-gram Language Model and a trigram Language Model. Both Mallet and Lingpipe provide a Java development API which were included in a test-bed evaluation application, along with TCASGL algorithm. With regards to the evaluation of the summarization and more specifically, the LSA systems, the JAMA, Java Matrix (Hicklin, Moler, Webb, Boisvert, Miller, & Remington) package was used.

### Corpus Profiles

The training and test dataset were randomly gathered from online Greek newspapers and was initially classified according to a unique classification scheme: Business and Finance, Culture, Health, Politics, Science and Technology, and Sports. The training corpus was comprised of 1015 articles and the test corpus of 353 articles for the classification. The corpora were randomly gathered by a number of newspapers in order to include a great number of authoring styles and vocabulary. The training corpus and test corpus were gathered over a period of a semester. They are both available at: [http://www.medialab.teicrete.gr/classification\\_corpus.rar](http://www.medialab.teicrete.gr/classification_corpus.rar). Each article in the training corpus underwent the stemming procedure and the resulting categories had the characteristics as depicted in Table 3.

Category	# of unique words	# of total words	Average word occurrence	Average weight
Business & Finance	5686	59777	10.51	0.000176
Culture	5750	26932	4.68	0.000174
Health	1181	3073	2.60	0.000847
Politics	7059	63218	8.96	0.000142
Science & Technology	3593	24429	6.80	0.000278
Sports	4696	15412	3.28	0.000213
Totals	27965	192841	6.139254653	0.000304909

Table 3. Nouns per category

As can be observed, each category is formed from around 3500 to 7100 unique words, except for Health class which is comprised of only 1181 unique nouns. The total number of words shows the total number of nouns included in each category (unique words times word frequency). Each category depicts its own average word occurrence and associated word weight. The average word weight ( $w_{i,j}$ ), excluding Health class, is around 0.000142 and 0.000278 depending on number of total words, while in Health class the average weight is 0.000847, denoting that a word present in Health class is 3 to 5 times more important than this word in any other domain.

TCASGL summarization module was evaluated and compared with a number of approaches as LEAD summarizer proposed in DUC 2002 that only considered the first 100 words of a document as a candidate summary, modified taking the first 30% of the sentences of the article as candidate summary, a baseline summarizer that extracted the first sentence of each document paragraph as candidate summary, Microsoft Summarizer available in Microsoft Office, a sample tf.idf-based algorithm, and, the LSA approach proposed by (Gong & Liu, 2001) both without and with stemming, and GreekSum (Pachantouris, 2005), an adaptation of the SweSum summarization engine for the Greek language, which is the only widely known available Greek summarizer. For the evaluation of the articles, both an automatic and a manual approach were used. The automatic approach used ROUGE-1 evaluation metrics, against which summarization engines in DUC conference in 2003 were evaluated. For this purpose, a test corpus from 237 Greek newspaper articles from online newspapers of varying content from various authors was gathered. Since both systems were developed primarily targeting the Greek language using Greek stemming techniques, the use of the DUC training sets seemed futile. Instead a corpus specific for the matter of the system evaluation was gathered. The test corpus was provided to a philologist who was asked to manually extract summaries of 30% from the original document resembling the evaluation procedure of the DUC conferences. She was also asked to provide notes and remarks on the evaluation approach she used as a general guideline upon the completion of the summarization tasks, which are also provided in the research results.



### GreekSum Summarization Engine

GreekSum (Pachantouris, 2005) is a Greek summarization engine based on modifications of SweSum engine originally developed for the Swedish language, developed by KTH/Stockholm, adapted for the Greek language. It is available at <http://www.nada.kth.se/iplab/hlt/greeksum/index.htm>. It is the only widely available Greek summarization engine. It supports supervised and unsupervised summarization. For the experiments, the trained version was used as, according to the author, it performs significantly better than the unsupervised version. SweSum utilizes statistical, location and template features in identifying important sentences. Such characteristics are the first sentence of a text segment, position characteristics (different per template used, while the templates are report and newspaper), machine and user-supplied keywords, word frequency and average sentence length. The engine implements a sentence extraction approach, while the segments are extracted using a linear weighted combination of the aforementioned features.

### tf.idf

A system was developed for comparison with TCASGL and GUTS that considers a normalized sentence weight, resulting from the average *tf.idf* score of each sentence term. *Tf.idf* was trained using a corpus composed of 1015 newspaper articles from various sources gathered over a period of six months –the same corpus used for the training of TCASGL system. The training corpus is available online at: [http://www.medialab.teicrete.gr/classification\\_corpus.rar](http://www.medialab.teicrete.gr/classification_corpus.rar)

### LSA

A system based on the fundamental work in LSA (Eq. 13) by (Gong & Liu, 2001) was developed as a test environment. The package used for the SVD was JAMA (Hicklin, Moler, Webb, Boisvert, Miller, & Remington). The algorithm proposed by the authors is based on the semantic representation of the SVD. More specifically, matrices  $\Sigma$  and  $V^T$  in SVD are used to denote the value of the topics discussed in the document and the contribution of the each sentence in each topic respectively. Thus, Gong and Liu propose acquiring from the  $V^T$ , the sentence that has the highest singular value

(column) for a given topic (singular index). Given that  $\Sigma$  is a diagonal matrix whose values are ordered descending, then it is safe to consider the top  $r$  topics from  $\Sigma$ , and thus consider the top  $r$  columns from  $V^T$ . Therefore, the top  $r$  most significant sentences are extracted, each one representing one topic, thus reducing redundancy. In this research, LSA was considered as proposed by Gong and Liu, while motivated by (Dumais, 2004), where the author stated that in small document sets stemming can improve performance, two LSA approaches were considered, one with and one without stemming.

### 7.5.2 GUTS Evaluation Approach

GUTS was tested against the same systems as the TCASGL on the same test corpus. However, instead of only using intrinsic evaluation through ROUGE-1, extrinsic evaluation was also used. Six members of the Multimedia Content Laboratory of the Technological Educational Institute of Crete consisting of graduate and post-graduate students and professors were asked to mark the summaries produced by the human summarizer, GreekSum and GUTS, in terms of coverage and cohesion. Each one was provided with an equal number of articles and their respective summaries in random order and was instructed to use a scale of 1-10 to evaluate each summary in terms of coverage – how well a summary represented the topics of an article.

## 7.6 TCASGL Evaluation Results

### 7.6.1 Classification Module

All classifiers were provided as input stems of words in the training step and stems of nouns in the sentences, in exactly the same manner. Thus, a number of interesting results were acquired. The complete results (Table 4) showed that TCASLG algorithm outperformed both Naive Bayes Classifier, 6-gram and trigram Language Model.

		<b>TCASLG</b>	<b>NBC</b>	<b>LM-6</b>	<b>LM-3</b>
Positive	#	326	302	284	294
	%	92,35	85,55	80,45	83,29
Negative	#	27	51	69	59
	%	7,65	14,45	19,55	16,71
Totals	#	353	353	353	353
	%	100	100	100	100

Table 4. Overall Classification Results

In Table 4, positive results are considered to be the ones where the algorithms managed to correctly match the human assigned class, whereas negative results are considered the ones that the algorithms failed to correctly identify. Thus, as may be seen, TCASGL outperformed both NBC and LM. More specifically, TCASGL classifier achieved a percentage of 92.35% correctly identified articles, while all other algorithms achieved well below 90%.

### Results per Category

In the following table (Table 5), the efficiency of the classifier per category is projected when compared to NBC and LMs, for single-labelled documents, as some interesting results may be extracted.

Categories		TCASLG		NBC		LM-6		LM-3	
		#	%	#	%	#	%	#	%
Business & Finance	Positive	66	95,65	63	91,30	59	85,51	60	86,96
	Negative	3	4,35	6	8,70	10	14,49	9	13,04
	Totals	69	100,00	69	100,00	69	100,00	69	100,00
Culture	Positive	54	96,43	54	96,43	53	94,64	52	92,86
	Negative	2	3,57	2	3,57	3	5,36	4	7,14
	Totals	56	100,00	56	100,00	56	100,00	56	100,00
Health	Positive	27	62,79	10	23,26	5	11,63	13	30,23
	Negative	16	37,21	33	76,74	38	88,37	30	69,77
	Totals	43	100,00	43	100,00	43	100,00	43	100,00
Politics	Positive	47	97,92	47	97,92	46	95,83	42	87,50
	Negative	1	2,08	1	2,08	2	4,17	6	12,50
	Totals	48	100,00	48	100,00	48	100,00	48	100,00
Science & Technology	Positive	73	93,59	70	89,74	62	79,49	69	88,46
	Negative	5	6,41	8	10,26	16	20,51	9	11,54
	Totals	78	100,00	78	100,00	78	100,00	78	100,00
Sports	Positive	59	100,00	58	98,31	59	100,00	58	9,31
	Negative	0	0,00	1	1,69	0	0,00	1	1,69
	Totals	59	100,00	59	100,00	59	100,00	59	100,00

Table 5. Classification Results per Category

First of all, the classifier produced for the given test corpus better (or as good results in some cases) as NBC and LM. The results between the algorithms are comparable

among all categories, except for Health class. Health class is made up of the smallest dictionary of all six categories, while it contains a number of words similar to Science & Technology class. TCASGL classification module correctly classifies 27 out of 43 Health articles in the test corpus (almost 63%), as opposed to NBC and LM which face serious problems (23%, 12% and 30% of the total Health articles correctly classified). A reason for that is that while NBC is indifferent to word co-occurrence it also treats every class as independent of one another trying to maximize the highest scoring category. This is the characteristic that tends to create bias towards a class with larger datasets (Rish, 2001). Health class in this test was comprised of the smallest dataset, sharing common words with Science and Technology, the latter being more than twice as big as Health corpus. Therefore, NBC tended to classify these documents incorrectly. Contrary to that, TCASGL was developed to treat each domain as equally probable, eliminating any bias. This was achieved through the weight calculation formula used to estimate word contribution to a category, through the observed frequency. Moreover, one of the preconditions in TCASGL classification module was that a word may exist in more than one category, with different weights per category. Therefore, it is important to estimate the overall importance of this word not only on one category but also on a cross-category level. These two estimates tend to produce less biased results on small datasets that share common words with larger datasets, such as Health class.

#### Tests with Ambiguous Data

During the tests, the classification produced category results that in some cases were ambiguous, most notably on articles that were manually classified by the newspapers into more than one class, since their content was semantically shared between 2 or 3 thematic areas. For example, the test article coded a83 originally classified as Politics by the newspaper, dealt with the cultural effects of the elections on a country through history. This implied that while Culture is the primary class for that article, Politics could also be a potential category. In fact TCASGL, identified both categories with Culture having a  $sf$  value of 0.2263 and Politics having a  $sf=0.2214$  while other classes'  $sf$  were in the region of 0.0067 to 0.1469. Motivated by that, an attempt to

verify how the system would operate with ambiguous input data was made, in order to check its robustness. 23 newspaper articles that could not be classified into one category were gathered. Two categories per article were manually assigned in order of similarity (e.g. Business and Finance/Politics denoting that this article fits into Business and Finance primarily and Politics secondarily) and the experiment was run again for these articles.

The results acquired are shown in Table 6.

		TCALSG
Positive	#	21
	%	91,30
Negative	#	0
	%	0,00
Ambiguous	#	2
	%	8,70
Totals	#	23
	%	100

Table 6. Classification with ambiguous input data

In this case, an extra selection apart from positive and negative is used, namely ambiguous. In this case, “positive” stands for correct classification of the document in all classes in order of classification similarity, “negative” failure to categorize the document in any of the classes it belongs to, and ambiguous either successfully categorizing into one of the categories of the document, but not all of them, or successfully categorizing into all categories but with different order from the one supplied by the human classifier. As can be observed, the efficiency of the classifier correctly identifying the classes of a document is similar to the one of single-label classification. Therefore, this hints that the system could be used for multi-label classification as well, since the results are very promising. However, this is beyond the scope of current work, as the intention of this research was to exclusively deal with single-label classification as part of the summarization procedure.

### 7.6.2 TCASGL Summarization Module Evaluation

Example: In this example a random summary produced by TCASGL with positional characteristics is presented. The article coded a14.txt is a sports article regarding

violence at football matches. It is important to point out that TCASGL classification module correctly identified the document as belonging to Sports.

Τα επεισόδια ξεκίνησαν όταν εκατό άτομα που φορούσαν ασπρόμαυρες φανέλες και διακριτικά του ΠΑΟΚ, ήρθαν σε σύγκρουση, όπως αναφέρουν οι άνθρωποι της διοίκησης του Άρη, με τους μεμονωμένους οπαδούς των "κιτρινόμαυρων", που βρέθηκαν στην Καβάλα για να παρακολουθήσουν το ματς.

Πολλά αυτοκίνητα έχουν υποστεί ζημιές, ενώ επίθεση δέχτηκε και το αυτοκίνητο του προέδρου της ΠΑΕ Άρη Θανάση Αθανασιάδη, ενώ και τα υπόλοιπα μέλη της ΠΑΕ προπηλακίστηκαν.

Φορώντας κράνη επιτέθηκαν σε ταβέρνες και μαγαζιά προκαλώντας αρκετές υλικές ζημιές. Οι ταραχοποιοί μπήκαν μέσα στο γήπεδο, στην διπλανή θύρα που είχαν πάρει εισιτήρια οι οπαδοί του Άρη, εκτοξεύοντας καπνογόνα και σπάζοντας καθίσματα.

Τα δακρυγόνα έπεφταν, επίσης, "βροχή" από την αστυνομία προς το πλήθος που προσπαθούσε να προσεγγίσει το γήπεδο. Στα επεισόδια αναμείχθηκαν και οπαδοί της Καβάλας που ήρθαν σε σύγκρουση με οπαδούς του Άρη.

Οι αστυνομικές δυνάμεις ήταν απούσες αφού δεν υπήρχαν μέτρα τήρησης της τάξης. Αντικείμενα δέχτηκαν οι παίκτες του Άρη κατά την είσοδο τους στον αγωνιστικό χώρο με αποτέλεσμα να επιστρέψουν στα αποδυτήρια.

Να σημειωθεί πως ορισμένοι οπαδοί του Άρη είχαν τυπώσει προσκλητήρια γάμου(!) για να μπορούν να ταξιδέψουν από τη Θεσσαλονίκη στην Καβάλα χωρίς να τους ενοχλήσει κανείς!

Λίγο πριν την έναρξη το "Ανθή Καραγιάννη" έχει 10.000 κόσμο, εκ των οποίων 2.500 οπαδοί είναι του Άρη.

Επεισόδια και στο ημίχρονο

Το γκολ που σημείωσε ο Κόκε στο 44ο λεπτό, δίνοντας το προβάδισμα με 0-1 στον Άρη, προκάλεσε νέα επεισόδια στο "Ανθή Καραγιάννη". Οπαδοί των "κιτρινών" μπήκαν στον αγωνιστικό χώρο για να πανηγυρίσουν μαζί με τους παίκτες του Άρη, ενώ με τη λήξη του ημιχρόνου εκτοξεύθηκαν φωτοβολίδες και καπνογόνα στον αγωνιστικό χώρο, αλλά και μεταξύ των οπαδών των δύο ομάδων.

Τα επεισόδια ξεκίνησαν όταν εκατό άτομα που φορούσαν ασπρόμαυρες φανέλες και διακριτικά του ΠΑΟΚ ήρθαν σε σύγκρουση όπως αναφέρουν οι άνθρωποι της διοίκησης του Άρη με τους μεμονωμένους οπαδούς των κιτρινόμαυρων που βρέθηκαν στην Καβάλα για να παρακολουθήσουν το ματς. Πολλά αυτοκίνητα έχουν υποστεί ζημιές ενώ επίθεση δέχτηκε και το αυτοκίνητο του προέδρου της ΠΑΕ Άρη Θανάση Αθανασιάδη ενώ και τα υπόλοιπα μέλη της ΠΑΕ προπηλακίστηκαν. Φορώντας κράνη επιτέθηκαν σε ταβέρνες και μαγαζιά προκαλώντας αρκετές υλικές ζημιές. Οι ταραχοποιοί μπήκαν μέσα στο γήπεδο στην διπλανή θύρα που είχαν πάρει εισιτήρια οι οπαδοί του Άρη εκτοξεύοντας καπνογόνα και σπάζοντας καθίσματα.

Figure 15. Original Document and TCASGL Summary -- code a14

After comparing the summaries extracted by each of the algorithms with the human summaries using ROUGE-1, the results depicted in Table 7 where extracted:

Method	ROUGE-1
TCASGL	0.4866
tf.idf	0.5462
LEAD	0.517
GreekSum	0.5589

Microsoft Summarizer	0.4666
Baseline	0.4965
LSA with Stemming	0.442
LSA without Stemming	0.447

Table 7 TCASGL without Positional Characteristics

The best performing algorithm is SweSum’s adaptation for the Greek language, followed by *tf.idf*. This is not strange considering that both algorithms are trainable and precondition a statistical or feature training analysis prior to the summarization process. The unexpected result was the performance of both LEAD and Baseline summarizers, since they only use an overly simplified approach when extracting their sentences. Moreover, the performance of the summarization through classification seems to only outperform Microsoft Summarizer. However, these results can be easily explained if one considers how the human summarization process is carried out (See Chapter 7.8) and the remarks by (Nenkova, 2005).

The test was run with TCASGL with positional characteristics. This yielded the following results:

Method	ROUGE-1
TCASGL	0.502
tf.idf	0.5462
LEAD	0.517
GreekSum	0.5589
Microsoft Summarizer	0.4666
Baseline	0.4965
LSA with Stemming	0.442
LSA without Stemming	0.447

Table 8. TCASGL without Positional Characteristics

As can be seen, the efficiency of the algorithm was slightly improved, surpassing in efficiency the baseline Baxendaliam summarization technique, yet it still lacks in efficiency, when compared to the other trainable summarizers.

### 7.6.3 TCASGL Efficiency

As can be seen from the results, TCASGL classification performs extremely well, surpassing in terms of efficiency other statistical classifiers, trained on the same document set and classes. In addition to that, it is obvious that TCASGL can perform

multi-label classification, as well, with equally good results. In terms of summarization, however, TCASGL fails to outperform any of the trainable summarizers included in the automatic evaluation (tf.idf, and GreekSum), while the lead summarization approach outperforms TCASGL, as well, for reasons analysed later in the thesis (See Chapters 7.8 and 8.2.3).

## 7.7 GUTS

### 7.7.1 GUTS Example

Prior to presenting the evaluation results of GUTS, a typical example summary is produced for article coded a14.txt (Figure 15) as in TCASGL (Figure 16). This example presents the summary produced by GUTS with location characteristics. As can be seen the system effectively produced a summary of roughly 30% the number of the original sentences.

Τα επεισόδια ξεκίνησαν όταν εκατό άτομα που φορούσαν ασπρόμαυρες φανέλες και διακριτικά του ΠΑΟΚ ήρθαν σε σύγκρουση όπως αναφέρουν οι άνθρωποι της διοίκησης του Άρη με τους μεμονωμένους οπαδούς των κιτρινόμαυρων που βρέθηκαν στην Καβάλα για να παρακολουθήσουν το ματς. Πολλά αυτοκίνητα έχουν υποστεί ζημιές ενώ επίθεση δέχτηκε και το αυτοκίνητο του προέδρου της ΠΑΕ Άρη Θανάση Αθανασιάδη ενώ και τα υπόλοιπα μέλη της ΠΑΕ προπηλακίστηκαν. Στα επεισόδια αναμειχθηκαν και οπαδοί της Καβάλας που ήρθαν σε σύγκρουση με οπαδούς του Άρη. Οι αστυνομικές δυνάμεις ήταν απούσες αφού δεν υπήρχαν μέτρα τήρησης της τάξης.

Figure 16. GUTS Summary – code a14

The corresponding similarity threshold for the original document can be approximated through Figure 17, where an analysis on the Jaccard similarity between each sentence is presented.

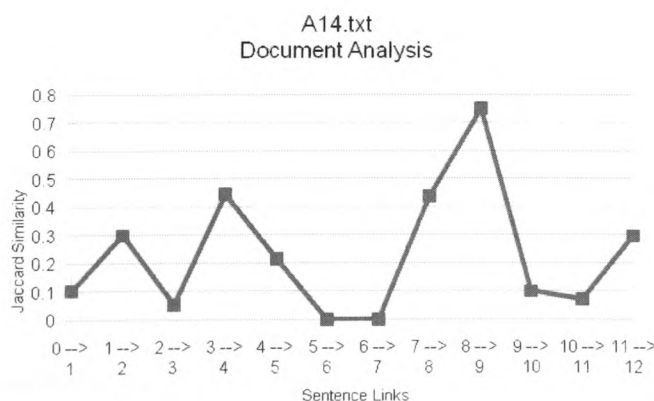


Figure 17. Jaccard Similarity - a14 document



As can be seen in Figure 17, the system has four local minima between sentences 2-3, 5-7 (Similarity Score 0) and sentences 10-11. The average of the four local minima is 0.031, and therefore the system identifies that topic shifts in the first step only occur between sentences 5-6 and 6-7 which are kept in the final extract. The sentences that were kept in the final extract are 0-1-5-6.

### 7.7.2 Automatic Evaluation of Results

The summaries from all eight systems (LEAD, baseline, *tf.idf*, MS Summarizer, GreekSum and GUTS without position characteristics, LSA with and without stemming) were extracted and compared against summaries extracted by a Greek philologist, assuming that human summaries are composed of the best possible sentences. The results acquired by the tests are depicted in Table 9:

Method	Average ROUGE-1 Score
LEAD	0,517
<i>tf.idf</i>	0,546
Baseline	0,496
MS Summarizer	0,467
<b>GreekSum</b>	<b>0,559</b>
GUTS	0,526
LSA with Stemming	0.442
LSA without Stemming	0.447

Table 9. GUTS without positional characteristics ROUGE-1 score

As can be seen, the best scoring system is the GreekSum. This is predictable since it is a semi-supervised algorithm with more information regarding semantic information available. GUTS scored better than any of the sample generic unsupervised algorithms, while its results are comparable to both the *tf.idf* and *GreekSum* supervised systems. Interesting remarks, however, on the efficiency of all the systems can be gleaned from the comments of the philologist (Chapter 7.8) regarding the human summarization process.

Using GUTS with position characteristics, setting a value of  $a=1.2$  in Eq. 29 yielded the results in Table 10.

Method	Average ROUGE-1 Score
LEAD	0,517
tf.idf	0,546
Baseline	0,496
MS Summarizer	0,467
<b>GreekSum</b>	<b>0,559</b>
GUTS with location characteristics	0,544
LSA with Stemming	0.442
LSA without Stemming	0.447

Table 10. GUTS with positional characteristics ROUGE-1 score

GUTS with location characteristics performs equally well as the sample tf.idf algorithm and slightly worse than GreekSum. However, keeping in mind that both tf.idf and GreekSum are supervised summarizers, in contrast to GUTS, its performance surpassed the initial expectations.

### 7.7.3 Manual Evaluation Results

The results of the manual summarization in terms of coverage and cohesion are presented in Table 11:

Algorithm	Result
GreekSum	6.268
GUTS	8.663
Human	10

Table 11. Human Evaluation Results Normalized

The results depict the average summarization score of each approach normalized by the highest scoring result. The results per document and the average results along with the test corpus are available at

[http://www.medialab.teicrete.gr:8080/summary\\_evaluation](http://www.medialab.teicrete.gr:8080/summary_evaluation).

A number of results can be extracted from this table. First of all, with regards to the GUTS algorithm, it is found to provide more consistent summaries than GreekSum. This is important, since the results do not correlate with automatic evaluation results as offered by ROUGE-1. On the other hand, the second important characteristic is the

human summaries extracted, which were considered to be the best among the three provided summaries, both in cohesion and in completeness. Thus, this implies that they could be used as gold standards in summary evaluation.

#### **7.7.4 GUTS Efficiency**

GUTS performs extremely well, as proven both in intrinsic evaluation through ROUGE-1 and manual extrinsic evaluation in terms of coverage and cohesion. More specifically, the system outperforms all knowledge-lean summarizers as LSA, baseline and LEAD, while it performs comparably to the tf.idf approach and GreekSum in terms of ROUGE-1 evaluation, especially when positional characteristics are taken into consideration. On the other hand, when considering the human evaluation GUTS, outperforms the highest scoring algorithm, in terms of cohesion and coverage as, shown by the manual evaluation of a mixed group of people with diverse educational backgrounds, over the document set of 237 articles.

#### **7.8 Human Summarization Approach**

In order to understand fully the evaluation results of both systems, a number of findings have to be presented regarding the approach in the summarization tasks of the human summarizer. The human summarizer is a philologist, PhD candidate in University of Crete, and was asked to comment on the approach taken to extract the summaries. These remarks have not been taken into consideration in any of the algorithms and are part of future work.

Her exact comments are: *"Prior to presenting our findings, it is important to point out that these are indicative findings rather than objective results, since language cannot be sealed and qualitatively evaluated using objective measures. This happens due to both the liquidity of the language, which is considered as the Message, and to other parameters that play an important role in its examination; the author-transmitter and the reader-receiver of the Message. Thus, the results are presented without any notion of absoluteness, framed with theory proof when possible."*

*The first characteristic we identified and included in our summaries is the first sentence of a paragraph. It is usual both in journalistic and in essay writing, that the context of the paragraph presented in the first sentence, the so-called thematic period.*

*The next element included in the summaries is the side heads that accompany the main document title. With regards to the latter I refer to the case where the text has side heads, while the paragraphs following them are omitted. In the case that the document has a numbered order, the element that we include in the summary is the first period that follows the numbered order, since this is the sentence that evaluates the meaning of those described before.*

*Apart from the elements included in the extract, equally important are the elements that were omitted from the summaries. Direct speech, for example, is an element that was omitted from the summaries in most cases. In journalist articles, direct speech is used to explain plainly, using the people involved as roles, what has been described earlier by the author. Thus, it is safe to conclude that omitting direct speech from the summary automatically includes the period preceding direct speech.*

*A second element that was left out of the summaries is punctuation marks that introduce induction along with the text they include or presage. Examples are parentheses, dashes, and “e.g.”. These three elements further analyze the thought of the author and offer details that are not necessary for the summary.*

*An ambiguous issue came up in the management of interviews, the problem being the rapid change in the two people talking. For two reasons we included the questions of the journalists rather than the answers; first of all, the questions included the potential response of the answer, while also, the words of the interviewee have already been included either from the title or from the side head of the document."*

### **7.9 Synopsis**

In this Chapter, a presentation of the available methodologies was made. Initially terms “intrinsic” and “extrinsic evaluation” are defined, followed by different evaluation approaches that have been proposed in the literature. This Chapter also discusses the

evaluation methodologies for both TCASGL and GUTS systems, while it provides an extensive analysis of both the evaluation results and remarks made by the human summarizer, whose summaries were as reference summaries in the evaluation.

The next chapter concludes this thesis, providing an overview of the accomplishments of research, with regards to research scope, aims and objectives. Finally it provides insight on future research that may enhance the efficiency of both systems

## 8. Conclusions and Future Work

*This chapter summarizes the research presented. Contributions to knowledge are also presented in relation to the initial aims and objectives listed in Chapter 1. In addition, it states the conclusions reached, as well as future work and final remarks.*

---

### 8.1 Introduction

At the start of the thesis, the aims and objectives of this research were underlined. More specifically, research revolved around the exploration of new summarization techniques based on the current state-of-the-art or novel approaches. In addition to that, research targeted on identifying whether linguistic information exploitation can assist in achieving better summary extraction. As a reference language for linguistic information research targeted on Greek, mostly because there is a limited availability on Greek summarization systems and methodologies. In order to facilitate these research aims the following research objectives were set:

- Develop an ML system and methodology to verify whether document summarization can be achieved through document classification. This was based on the assumption that domain knowledge is important for summary extraction, hence identifying the domain could actually assist in document summarization. As a side research objective, the development of a novel classification scheme was set, after initial research in the area of statistical classifiers, in order to better suit our methodology.
- Develop an NLP system that would use already researched and novel features in sentence extraction for document summarization. Especially, the system should try to exploit linguistic characteristics of the reference language, while at the same time not limit itself to the reference language.

## 8.2 Overview of Research

Following the research objectives 4 areas of interest were identified:

- The identification of important document words (noun-adjectives) through grammatical analysis, and specifically stemming.
- Document classification through a statistical trainable summarizer
- Document summarization through classification
- Document summarization based on NLP

### 8.2.1 Stemming

Stemming was a task carried out as an adjunct to document summarization (both ML and NLP). Through stemming, the first linguistic rules of Greek language were set. More specifically, one of the problems of linguistic analysis is to identify important words. It has been suggested by (Bouras & Tsogkas, 2010) that nouns hold important meanings regarding the subjects and the objects of a sentence, which agrees with the initial considerations of the thesis author. The available approaches involve grammatical or syntactical analysis. Both approaches require modeling of specific linguistic characteristics. In essence, research in this area resulted in a grammatical set of rules for noun identification that was used for both TCASGL and GUTS.

### 8.2.2 Document Classification

Document classification is an integral part of Machine Learning systems. TCASGL was based on the assumption that domain knowledge can assist document summarization. However, the available techniques were either quite complex or restrictive and less efficient. Therefore, as a side research objective document classification was set, with the goal to outperform other statistical classifiers within the scope of document summarization. The approach selected was roughly based on the same rules of statistical independence known as Bayes rule, yet it differed a lot on the fact that it tried to eliminate statistical independence by considering the contribution of the input document words to all of the categories, rather than forming a strict selection model. This enabled a less strict classification algorithm that suited

both single label and multi label classification. The outcome of this research was used on TCASGL for the document classification module.

### **8.2.3 Machine Learning Summarization**

Machine Learning algorithms have been used in the past for document summarization. However, all approaches in literature considered ML as a sentence selection feature, rather than a way of extracting terminological information from documents. The first researchers to imply such a use for an ML system were (Barzilay & Lee, 2004), where they proposed a probabilistic model for document summarization based on domain knowledge. This influenced the current research, thus orienting part of it to further explore if document classification can assist document summarization. Through that, a new approach emerged on document summarization fulfilling one of the aims set initially.

### **8.2.4 Natural Language Processing Summarization**

NLP is primarily the area of interest when researchers refer to document summarization. However, very little research has been done to exploit linguistic characteristics for Greek language, most of the researchers targeting on English and their native language. Based on that, research targeted on finding a way to model the Greek linguistic characteristics through the stemming module, but also be flexible enough to facilitate any language. Moreover, a new heuristic selection feature named “conceptual flow” was introduced, that helped identify the different topic clusters of the document and evaluate topic shift and topic regression. This methodology fulfilled the aim of producing a methodology for summary extraction initially targeting for the Greek language, while in the end of research finding out that it could facilitate other languages, as well.

## **8.3 Overview of TCASGL Methodology**

TCASGL is a ML system based on document summarization through classification. The system is composed of three modules:



- A training module used for training the classifier into forming the classes and feature weights.
- A classification module, where the system classifies the document into one of the available classes.
- A summarization module used to identify and extract the most important sentences, according to the class identified by the classifier.

In the training phase TCASGL, the system estimates the importance of each word and its contribution to each class by considering the number of its occurrences in the class, the number of its occurrence in other classes and the length of each available class. The training phase utilizes statistics in extracting the word class weight, minimizing the extent of effect of the statistical independence of the features, as implied by the Naïve Bayes Assumption upon which TCASGL is based.

During the classification phase, TCASGL classifies a document in one (or more than one) of the available classes. Classification utilizes the average of word weights per class, as it has been computed by the training phase. The decision of the class defines the reference dictionary that is going to be used in the summarization phase.

The summarization phase of TCASGL, calculates the importance of document sentences based on the words it is composed of. For each sentence, the system computes the normalized average of sentence terms for the class the document belongs to and maintains the topmost  $x$  important sentences in order of appearance in the original document. An extension to the summarization includes, apart from the class weight, sentence/paragraph position as an important feature in estimating sentence importance, producing better results than the main TCASGL methodology.

Setting aside the evaluation of TCASGL, it can be said that it fulfilled the objectives set during this research. A thorough research on whether document classification can assist document summarization was undertaken, with important results both in the area of document classification and in the area of document summarization. The approach of identifying topics and classifying the documents according to a set of predefined categories, while did not yield extremely good results in summarization,

leaves room for improvement. Moreover, the classification scheme proved to outperform other statistical classification algorithms.

#### **8.4 Overview of GUTS Methodology**

GUTS is an NLP system that exploits several document-specific characteristics in identifying and extracting important sentences. The main idea behind GUTS is to exploit the “conceptual flow” of the document – how authors organize their sentences in order to fully describe a topic. According to “conceptual flow”, GUTS estimates the potential topic shifts that may occur between consecutive document sentences. This is achieved by forming a term by term matrix – abstract semantic lexicon – denoting the observed term co-occurrence with regards to the expected term co-occurrence. The system assigns a bias weight if two terms are found to co-occur more times than expected in the document. In this first summarization step, the system calculates the average similarity of sentences, through a modified Jaccard similarity measure, that considers word co-occurrence bias in addition to similar words. This enables an initial sentence clustering on consecutive sentences, by identifying the local sentence similarity minima on consecutive sentences and comparing them to the average sentence similarity computed for the document. The most important sentences of each cluster are extracted and an initial summary is formed. In case this summary is greater than the expected length of summary, a similar approach is considered on non consecutive sentences. This step assists in determining topic re-visiting in the document, thus extracting more complete clusters of sentences. Again, the most important sentences in each cluster are extracted and form a second summary. The last step of the algorithm, utilized only in ensuring the desired length of summary, ignores the abstract semantic relation of terms and topic clusters, and extracts sentences according to term occurrence. An extension to the algorithm has also been proposed, by including the relative sentence position in a paragraph, producing better results than the main GUTS methodology.

GUTS manages to include a novel feature as conceptual flow, and also take advantage of some of the linguistic characteristics of Greek language. The aim of modeling completely Greek language proved to be overoptimistic. However, some of the Greek

grammatical rules were used in the algorithm. In addition to that, the use of a dedicated Greek stemmer for the identification of important information, regardless of some obvious shortcomings, in general behaved pretty well in the summarization. More essentially, GUTS was comparable to other algorithms in terms of quantitative characteristics (ROUGE-1), and better in terms of qualitative characteristics as proven by human evaluation of the extracted summaries.

### **8.5 Contribution to Knowledge**

Document classification and summarization as tasks are neither new, nor poorly researched. A number of approaches are documented for both tasks in the literature, reporting excellent results in most cases. However, as it has been shown by the research presented in this thesis, improvements can be made, either based on manipulating the fundamentals of these approaches, or by considering completely new techniques.

A minor improvement on what has already been considered in the literature, presented in this research, is the use of stemming as a means of identifying parts of speech, instead of utilizing a POS tagger. Stemming has already been considered in work by Porter (1980) and Kalamboukis (1995), yet it was always considered as a general task in text retrieval and never with specific adaptations to assist POS tagging.

With regards to document classification, both NBC and LM have been used in statistical classification of documents, with very good results. In this research, a slightly different approach was considered outperforming both statistical classifiers in terms of correctly identified document classes, on the same document set. The classifier is based on the same foundations of the NBC (naive assumption of statistical independence of words), yet instead of utilizing the product approach considered in both NBC and LM, it uses a sum approach offering smoother error propagation and multi-label capabilities.

Document summarization through classification in the current form seems to offer worse results than other approaches. On the contrary, the knowledge-lean approach proposed in this thesis under the name of GUTS, significantly outperforms the LSA

approach considered in (Gong & Liu, 2001). It is also important to point out that the algorithm outperforms all knowledge-lean approaches, yields comparable results to the *tf.idf* approach, while falls very little behind GreekSum in terms of ROUGE-1 evaluation. Extrinsic evaluation, however, has proven that the GUTS system provides more coherent summaries, covering more document information, in approximately the same length of summary.

### 8.6 Limitations

The following limitations have been identified in the algorithms and systems:

#### TCASGL

- While the classification algorithm allows multi-label classification, the summarization algorithm only considers the word weight on the selected category disregarding the importance a word might have on other categories. More specifically, during the evaluation of a sentence, a normalized category weight is taken into account for the selected category and not for the rest of the categories.
- The system is highly dependent on the input categories and more specifically the lack of a high number of sets. It is yet to be estimated how it would work on a greater number of potentially overlapping datasets.
- TCASGL is trained according to newspaper articles, and therefore in its current state can only perform efficiently newspaper summarization tasks.

#### GUTS

- GUTS, with the current set of selection features, cannot be easily modified to accommodate multi-document summarization. More specifically, conceptual flow is a single-document summarization selection feature as it considers the flow of authoring within a document. Extending it to multi-document summarization without a complementary multi-document summarization feature makes little sense.

- GUTS heavily depends on the input language. It has been developed with Greek in mind and follows the Greek grammatical rules. Thus, adaptations or extensions to other languages imply an extensive linguistic analysis for the rules to be used, since a simple stop-word dictionary addition to identify common words is less than efficient.

#### Both methodologies

- TCASGL and GUTS are based on the fact that nouns are important in identifying the meaning of a word, by trying to isolate noun stems. This approach, however, is limited by the potential of the stemming algorithm. Unfortunately in Greek language nouns and adjectives share a lot of suffixes, hence introducing noise in the noun extraction approach.

### **8.7 Future Work**

While the results presented here surpass the author's expectation, research provided insight on a number of areas that can be investigated with the aim to achieve better results. These areas either deal with the basis of each algorithm as the stemming or POS tagging or with the summarization algorithms and what they consider as important selection features.

More specifically, the following have been identified as areas that can be investigated in order to achieve optimal results:

- Summarization can be assisted if other parts of speech could be included for consideration in the algorithm, and especially verbs. Verb are used to denote the transfer of action between the sentence subject and the object, or used to denote the effect of an action when used in a passive form. Their identification is considered in the stemming module provided, but due to Greek language particularities, especially in verbs (See Table 1. Greek Verbs), their use through stemming is very difficult in current form. This could be tackled with a POS tagger as it would provide a more definite identification criterion than simple stemming, provided the problems that were initially identified are tackled.

- ML summarization through TCASGL offers average results. This is due to the small number of features considered in extracting the summaries. TCASGL uses term class weight, term density (average term weight of a sentence) and relative paragraph-sentence position as features in selecting most significant sentences. While extrinsic evaluation was not performed in the case of TCASGL, ROUGE-1 scores, position it as an average system with the current feature set. Thus, the inclusion of surplus feature sets, as the ones provided in the literature as named entities, title terms, keywords, and cue phrases, will probably enhance the system's efficiency.

Other potential research on the area includes ideas that have been proposed in the literature that could be used in extending the algorithms' usage:

- The classifier could be tested against multi-label classification algorithms in terms of efficiency and be applied in a great variety of ML applications, as e.g. plagiarism detection or image classification.
- The GUTS summarizer currently supports Greek language text as the stemmer and lemmatizer uses a set of Greek language stop-list and Greek noun endings. The inclusion of verbs in the current module seems problematic. However, the system is built around generic multilingual approaches, and thus a set of different stop-lists and POS tagging, can easily lead to the extension of GUTS system in multiple languages.

### **8.8 Final Remarks**

The systems presented utilize a number of novel and already researched techniques in acquiring general content results. Their efficiency has been demonstrated using examples over different training and test data sets, while both human and automatic evaluation was considered. The systems have provided important results both in terms of statistical classification and summarization. However, efficient results can only be provided through extensive real-time use, or extended usability and added value in different ML and NLP tasks.

---

## References

- Alias-I. (n.d.). *Lingpipe*. Retrieved 7 20, 2010, from Lingpipe: <http://alias-i.com/lingpipe>
- Amini, M., & Gallinari, P. (2002). The use of unlabeled data to improve supervised learning for text summarization. *ACM-SIGIR02*, (pp. 105-112). Tampere, Finland.
- Amini, M., Usunier, N., & Gallinari, P. (2005). Automatic Text Summarization based on word-clusters and ranking algorithms. *Lecture Notes in Computer Science* , 3408, 142-156.
- Angheluta, R., De Busser, R., & Moens, M. (2002). The Use of Topic Segmentation for Automatic Summarization. *Workshop on Text Summarization in Conjunction with the ACL 2002 and including the DARPA/NIST sponsored DUC 2002 Meeting on Text Summarization*, (pp. 11-16). Pennsylvania, USA.
- Aone, C., Okurowski, M., Gorinsky, J., & Larsen, B. (1997). A scalable summarization system using NLP. *Proceedings of the ACL'97 EACL'97 Workshop on Intelligent Scalable Text Summarization*, (pp. 66-73). Madrid, Spain.
- Banko, M., Mittal, V., Kantrowicz, M., & Goldstein, J. (1999). Generating extraction-based summaries formhand-written summaries by aligning text-spans. *PACLING-99*. Waterloo, Ontario, Canada.
- Barzilay, R., & Elhadad, M. (1997). Using Lexical chains for Text Summarization. *ACL Workshop on Intelligent Scalable Text Summarization*, (pp. 10-17). Madrid, Spain.
- Barzilay, R., & Lee, L. (2004). Catching The Drift: Probabilistic content models with application to summrization and generation. *HLT-NAACL 04*, (pp. 113-120). Boston, USA.
- Baxendale, P. (1958). Machine-made index for technical literature: an experiment. *IBM Journal* , 354-361.

---

Bellare, K., das Sarma, A., das Sarma, A., Loiwal, N., Mehta, V., Ramakrishnan, G., & Bhattacharyya, P. (2004). Generic text summarization using WordNet. *International Conference on Language Resources and Evaluation*.

Bhandari, H., Shimbo, M., Ito, T., & Matsumoto, Y. (2008). Generic Text Summarization Using Probabilistic Latent Semantic Indexing. *3rd International Joint Conference on Natural Language Processing - IJCNLP*, (pp. 133-140). Hyderabad, India.

Bosma, W. (2005). Query-Based Summarization using Rhetorical Structure Theory. *15th Meeting of CLIN*, (pp. 29-44). Leiden, Netherlands.

Bouras, C., & Tsogkas, V. (2010). Improving Text Summarization Using Noun Retrieval Techniques. *Lecture Notes in Computer Science*, 5178, 593-600.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, Diversity-Based Reranking for reordering documents and producing summaries. *SIGIR'98*. Melbourne, Australia.

Conroy, J. M., Schlesinger, J., Goldstein, J., & O'Leary, D. (2004). Left-Brain/Right-Brain Multi-Document Summarization. *4th Document Understanding Conference (DUC'04)*. Boston, USA.

Conroy, J., & O'Leary, D. (2001). *Text summarization via hidden markov models and pivoted QR matrix decomposition*. Technical Report, University of Maryland, Maryland, USA.

Croft, W. (2003). Language models for Information Retrieval. *19th International Conference on Data Engineering*, (pp. 3-7). Bangalore, India.

Dai, W., Xue, G. R., Yang, Y., & Yu, Y. (2007). Transferring Naive Bayes Classifiers for Text Classification. *22nd AAAI Conference on Artificial Intelligence* (pp. 540-545). Vancouver, Canada: AAAI Press.

Deerwester, S., Dumais, S., Furnas, G., Harshman, R., Landauer, T., Landauer, K., & Streeter, L. (1988). *Patent No. Computer information retrieval using latent semantic structure*. USA.



---

Dorr, B., Monz, C., President, S., Schwartz, R., & Zajic, D. (2005). A Methodology for Extrinsic Evaluation of Text Summarization. Does ROUGE Correlate? *ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (pp. 1-8). Sydney, Australia.

Dumais, S. (2004). Latent Semantic Analysis. *Annual Review of Information Science and Technology* , 38 (1), 188-230.

Edmundson, H. (1969). New methods in automatic extracting. *Journal of the Association of Computer Machinery* , 16 (2), 264-285.

Erkan, G., & Radev, D. (2004a). LexPageRank: Prestige in Multi-Document Text Summarization. *Proceedings of EMNLP 2004* (pp. 365-371). Barcelona, Spain: Association for Computational Linguistics.

Erkan, G., & Radev, D. (2004b). LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization. *Journal of Artificial Intelligence Research* , 22, 457-479.

Fattah, M., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech and Language* , 23, 126-144.

Filatova, E., & Hatzivassiloglou, V. (2004). Event-based extractive summarisation. *ACL-2004*, (pp. 104-111). Barcelona, Spain.

Goldstein, J., Mittal, V., Carbonell, J., & Kantrowicz, M. (2000). Multi-Document Summarization By Sentence Extraction. *2000 NAACL-ANLP Workshop on Automatic Summarization* (pp. 40-48). Seattle, USA: Association for Computational Linguistics.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. *24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR'01*, (pp. 19-25). New Orleans, USA.

---

Greek POS Tagger, A. U. (n.d.). *Natural Language Processing Group*. Retrieved 12 5, 2010, from [http://nlp.cs.aueb.gr/software\\_and\\_datasets/AUEB\\_Greek\\_POS\\_tagger.tar.gz](http://nlp.cs.aueb.gr/software_and_datasets/AUEB_Greek_POS_tagger.tar.gz)

Grishman, R., & Sundheim, B. (1996). Message Understanding Conference - 6: A Brief History. *Proceedings of the 16th conference on Computational linguistics - COLING96*, (pp. 466-471). Copenhagen, Denmark.

Harnly, A., Nenkova, A., Passonneau, R., & Rambow, O. (2005). Automation of Summary Evaluation by the Pyramid Method. *International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. Borovets, Bulgaria.

Hicklin, J., Moler, C., Webb, P., Boisvert, R., Miller, B. R., & Remington, K. (n.d.). *JAMA: Java matrix package*. Retrieved 7 10, 2011, from JAMA: Java matrix package: <http://math.nist.gov/javanumerics/jama>

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C. (2000). Question Answering in Webclopedia. *9th Text REtrieval Conference - TREC-9*, (pp. 655-664). Maryland, USA.

Hovy, E., Lin, C., Zhou, L., & Fukumoto, J. (2006). Automated Summarization Evaluation with Basic Elements. *5th Conference on Language Resources and Evaluation - LREC*. Genoa, Italy.

Jaoua, M., & Ben Hamadou, A. (2003). Automatic Text Summarization on scientific articles based on Classification and Extract's Popoulation. *Lecture Notes in Computer Science* , 2588, 623-644.

Java. (n.d.). Retrieved from Java: <http://www.java.com>

Jing, H., Barzilay, R., Mckeown, K., & Elhadad, M. (1998). Summarization Evaluation Methods: Experiments and Analysis. *AAAI Symposium on Intelligent Summarization*, (pp. 60-68). Palo Alto, USA.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* , 28 (1), 11-21.

---

Kalamboukis, T. (1995). Suffix stripping with modern greek. *Program: electronic library and information systems* , 29 (3), 313-321.

Kantrowicz, M. (1992). *Term-length term-frequency method for measuring document similarity and classifying text*. Patent, USA.

Karanikolas, N., & Skourlas, C. (2010). A Parametric Methodology for Text Classification. *Journal of Information Science* , 36 (4), 421-442.

Konstantis, S., & Pintelas, P. (2004). Increasing the Classification Accuracy of A Single Bayesian Classifier. *Lecture Notes in Artificial Intelligence* , 3192, 198-207.

Kruengkrai, C., & Jaruskulchai, C. (2003). Generic text summarization using local and global properties of sentences. *IEEE/WIC International Conference on Web Intelligence (WI'03)*, (pp. 201 - 206 ). Halifax, Canada.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. *18th Annual International ACM/SIGIR Conference on Research and Development in IR*, (pp. 66-73). Seattle, USA.

Landauer, T., & Dumais, S. (n.d.). *Latent Semantic Analysis*. Retrieved 6 6, 2011, from Scholarpedia: [http://www.scholarpedia.org/article/Latent\\_semantic\\_analysis](http://www.scholarpedia.org/article/Latent_semantic_analysis)

Lee, J., Park, s., Ahn, c., & Kim, D. (2009). Automatic generic document summarization based on non-negative matrix factorization. *Information Processing and Management* , 45, 20-34.

Li, L., Zhou, K. G., Zha, H., & Yu, Y. (2009). Enhancing Diversity, Coverage and Balance for Summarization through Structure Learning. *WWW 2009*, (pp. 71-80). Madrid, Spain.

Lin, C., & Hovy, E. (2002). Automated Multi-document Summarization in NeATS. *Second international conference on Human Language Technology Research, HLT02* (pp. 59-62). San Diego, USA: Morgan Kaufmann Publishers Inc.

- 
- Lin, C., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. *Language Technology Conference (HLT-NAACL 2003)*. Edmonton, Canada.
- Lin, C., & Hovy, E. (1997). Identifying topics by position. *Applied Natural Language Processing Conference*, (pp. 283-290). Washington, USA.
- Lin, C.-Y., & Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. *COLING 2000*, (pp. 495-501). Strasbourg, France.
- Liu, F., & Liu, Y. (2008). Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries. *ACL-08: HLT, Short Papers (Companion Volume)*, (pp. 201-204). Columbus, USA.
- Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal* , 159-165.
- Mani, I., & Bloedorn, E. (1997). Multi-document Summarization by Graph Search and Matching. *CoRR* , *cmp-lg/9712004*.
- Mann, W., & Thompson, W. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text* , 8 (3), 243-281.
- Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction To Informaiton Retrieval*. Cambridge University Press.
- Marcu, D. (1997). From discourse structure to text summaries. *35th Annual Meeting of the Association for Computational Linguistics - ACL97*, (pp. 82-88). Madrid, Spain.
- McCallum, A. (n.d.). *MALLET: A Machine Learning for Language Toolkit*. Retrieved 7 20, 2010, from <http://mallet.cs.umass.edu>
- McCallum, A., & Nigam, K. (1998). A comparison of Event Models for Naive Bayes Text Classification. *AAAI-98 Workshop on Learning for Text Categorization*, (pp. 41-48). Wisconsin, USA.
-

---

McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Schiffman, B., & Teufel, S. (2001). Columbia Multi-Document Summarization: Approach and Evaluation. *Document Understanding Conference (DUC01)*. New Orleans, USA.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order to Text. *EMNLP'2004*, (pp. 404-411). Barcelona, Spain.

Miller, G. (1995). Wordnet: A lexical Database for English. *Communications of the ACM*, 38 (11).

Moens, M., Angheluta, R., & Dumortier, J. (2005). Generic technologies for single- and multi-document summarization. *Information Processing and Management*, 41, 569-586.

*MySQL Community Edition*. (n.d.). Retrieved from MySQL Community Edition: [www.mysql.com](http://www.mysql.com)

Nenkova, A. (2005). Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference. *20th National Conference on Artificial Intelligence (AAAI 2005)*. Pittsburg, USA.

*Netbeans*. (n.d.). Retrieved from Netbeans: [www.netbeans.org](http://www.netbeans.org)

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 23 (2-3), 103-134.

Nobata, C., Sekine, S., Uchimoto, K., & Isahara, H. (2002). A summarization system with categorization of document sets. *3rd NTCIR-TSC Workshop*, (pp. 33-38). Tokyo, Japan.

Nomoto, T., & Matsumoto, Y. (2001). A new approach to unsupervised text summarization. *SIGIR-01*, (pp. 26-34). New Orleans, USA.

Over, P., Dang, H., & Harman, D. (2007). DUC in Context. *Information and Management Processing*, 43, 1506-1520.

---

Pachantouris, G. (2005). *GreekSum - A Greek Text Summarizer*. Master Thesis, KTH-Stokholm University.

Paice, C., & Jones, P. (1993). The identification of important concepts in highly structured technical papers. *16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR93* (pp. 69-78). Pittsburg, USA: ACM.

Peng, F., Schuurmans, D., & Wang, S. (2004). Augmenting Naive Bayes Classifiers with Statistical Language Models. *Information Retrieval* , 7 (3-4), 317-345.

Ponte, J. M., & Croft, W. B. (1998). A language modelling approach to information retrieval. *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 275-281). Melbourne, Australia: ACM.

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

Radev, D., Fan, W., & Zhang, Z. (2001). WebInEssence: A Personalized Web-Based Multi-Document Summarization and Recommendation System. *NAACLWorkshop on Automatic Summarization*. Pittsburgh, USA.

Radev, D., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. *ANLP/NAACL Workshop on Summarization*, (pp. 21-30). Seattle, USA.

Rennie, J. D., Shih, L., & Karger, D. (2003). Tackling the Poor Assumptions of the Naive Bayes Classifier. *20th International Conference on Machine Learning (ICML-2003)*, (pp. 616–623). Washington DC, USA.

*Reuters Corpus*. (n.d.). Retrieved 7 30, 2010, from Reuters: <http://trec.nist.gov/data/reuters/reuters.html>

Ripley, B. D. (1994). Neural Networks and Related Methods for Classification. *Journal of the Royal Statistical Society. Series B (Methodological)* , 56 (3), 409-456.

- 
- Rish, I. (2001). An Empirical Study of the Naive Bayes Classifier. *IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, (pp. 41-46). Seattle, USA.
- Rocchio, J. (1971). Relevance Feedback in Information Retrieval. *SMART Retrieval System, Experiments in Automatic Document Processing* , 313-323.
- Saggion, H., Bontcheva, K., & Cunningham, H. (2003). Robust Generic and Query-Based Summarization. *10th conference on European chapter of the Association for Computational Linguistics*. Budapest, Hungary.
- Scott, S., & Mattwin, S. (1999). Feature Engineering for Text Classification. *16th International Conference on Machine Learning* (pp. 379-388). Bled, Slovenia: Morgan Kauffman Publishers.
- Seki, Y. (2002). Sentence extraction by tf/idf and position weighting from newspaper articles. *3rd NTCIR-TSC Workshop*. Tokyo, Japan.
- Shen, D., Sun, J., Li, H., Yang, Q., & Chen, Z. (2007). Document Summarization using conditional random fields. *20th international joint conference on Artificial Intelligence, IJCAI'07*. Hyderabad, India.
- Song, Y., Han, K., & Rim, H. (2004). A Term Weighting Method Based on Lexical Chain for Automatic Summarization. *Lecture Notes in Computer Science* , 2945, 636-639.
- Sparck Jones, K. (2007). Automatic Summarizing: The State of The Art. *Information Processing and Management* , 43 (6), 1449-1481.
- Srikanth, M., & Srihari, R. (2002). Biterm language models for document retrieval. *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 425-426). Tampere, Finland.
- Steinberger, J., & Jezek, K. (2005). Text summarization and Singular Value Decomposition. *Lecture Notes in Computer Science* , 3261, 245-254.
-

---

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *7th International Conference on Information Systems Implementation and Modelling, ISIM'04*, (pp. 93-100). Roznov pod Radhostem, Czech Republic.

Strzalkowski, T., Wang, J., & Wise, B. (1998). *A robust practical text summarizer*. AAAI.

Svore, K., Vanderwende, L., & Burges, C. (2007). Enhancing Single-document Summarization by Combining RankNet and Third-party Sources. *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (pp. 448-457). Prague, Czech Republic.

*TREC Corpus*. (n.d.). Retrieved 7 30, 2010, from TREC: <http://trec.nist.gov>

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2006). A Review of Multilabel Classification Methods. *2nd ADBIS Workshop on Data Mining and Knowledge Discovery*, (pp. 99-109). Thessaloniki, Greece.

Wan, X., Yang, J., & Xiao, J. (2007). Manifold-Ranking Based Topic-Focused Multi-Document Summarization. *International Joint Conference on Artificial Intelligence, IJCAI2007*, (pp. 2903-2908). Hyderabad, India.

Wang, M., Li, C., & Wang, X. (2007). Chinese automatic summarization based on thematic sentence discovery. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, (pp. 482-486). Shandong, China.

Yeh, J. Y., Ke, H. R., Yang, H. W., & Meng, I. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management*, 41, 75-95.

Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'02*, (pp. 113-120). Tampere, Finland.



## APPENDIX A – Noun and Verb Endings and Stop-Lists

### Noun and Verb Endings

<b>Noun Endings</b>	"ας", "α", "αδες", "αδων", "ες", "ων", "ης", "η", "ηδες", "ηδων", "ες", "ων", "ες", "εδες", "εδων", "ους", "ουδες", "ουδων", "ος", "ου", "ο", "ε", "οι", "ων", "ους", "εας", "εα", "εις", "εων", "α", "ας", "ων", "ες", "αδες", "αδων", "η", "ης", "ων", "ες", "εις", "εων", "ω", "ως", "ος", "ου", "ο", "οι", "ων", "ους", "ο", "ου", "α", "ων", "ατα", "ατων", "ι", "ιου", "ια", "ιων", "υ", "ιου", "ια", "ιων", "ος", "ους", "η", "ων", "α", "ατος", "ατα", "ατων", "ας", "ατος", "ατα", "ατων", "ως", "ωτος", "ατα", "των", "ον", "οντος", "οντα", "οντων", "αν", "αντος", "αντα", "αντων", "εν", "εντα", "εντος", "εντων"
<b>Verb Endings</b>	"ω", "εις", "ει", "με", "τε", "ουν", "οντας", "ωντας", "μαι", "σαι", "ται", "αν", "ηκα", "ηκε", "ναι", "ε"

### Stop-List

<b>Articles</b>	"ο", "του", "στου", "το", "τον", "στον", "στο", "η", "της", "στης", "τη", "την", "στην", "στη", "οι", "των", "στων", "τους", "στους", "τις", "στις", "τα", "στα", "ενας", "ενος", "ενα", "μια", "μιας", "μια", "μιας", "του"
<b>Conjunctions</b>	"και", "κι", "ουτε", "μητε", "ουδε", "μηδε", "η", "ειτε", "μα", "αλλα", "παρα", "ομως", "ωστοσο", "ενω", "αν", "μολονοτι", "μονο", "λοιπον", "ωστε", "αρα", "επομενως", "που", "θα", "απο", "δηλαδη", "πως", "οτι", "οταν", "εαν", "καθως", "αφου", "αφοτου", "πριν", "μολις", "προτου", "ωσπου", "ωσοτου", "οσο", "οποτε", "γιατι", "επειδη", «αμα», «να», «μη», «μην», «μηπως», «για», «ας», «δεν», «οχι», «ναι», «οπου», «πλεον»
<b>Interjections</b>	"α", "μπα", "χμ", "ποπο", "ω", "αχ", "οχου", "αου", "αλι", "οχ", "αλιμονο", "ε", "ου", "αχαχουχα", "ειθε", "μακαρι", "αμποτε", "ευγε", "μπραβο", "ουφ", "πουφ", "πα πα πα", "αντε", "αμε", "μαρς", "αλτ", "στοπ", "σουτ", "στοπ", "αερα"
<b>Prepositions</b>	με", "συν", "σε", "πλην", "για", "επι", "ως", "δια", "προς", "μειον", "κατα", "υπερ", "μετα", "περι", "παρα", "εν", "αντι", "εκ", "απο", "εξ", "διχως", "υπο", "χωρις", "ισαμε"
<b>Pronouns</b>	"εγω", "αυτοι", "εμενα", "αυτους", "εμεις", "αυτη", "αυτην", "εμας", "αυτης", "μου", "αυτες", "με", "τος", "μας", "του", "εσυ", "τον", "εσενα", "τα", "σου", "τους", "σε", "τη", "εσεις", "της", "εσας", "τη", "την", "σας", "το", "αυτος", "τες", "αυτου", "τις", "αυτο", "αυτον", "αυτων", "καποιος", "καποιου", "καποια", "καποιο", "εκεινος", "εκεινους", "εκεινη", "εκεινης", "εκεινο", "εκεινοι", "εκεινων", "εκεινες", "εκεινου", "εκεινα", "καθε", "οποιος",

	"οποια", "οποιο", "τετοιος", "τετοια", "τετοιου",
<b>Adverbs</b>	"που", "φετος", "καπου", "πως", "πουθενα", "καπως", "εδω", "αλλιως", "εκει", "ετσι", "αυτου", "μαζι", "παντου", "οπως", "οπου", "καθως", "οπουδηποτε", "ως", "πανω", "σαν", "κατω", "διαρκως", "καταγης", "μεμιας", "μεσα", "μονομιας", "εξω", "επισης", "εμπρος", "μπρος", "ιδιως", "πισω", "κυριως", "απεναντι", "ειδεμη", "γυρω", "τυχον", "ολογυρα", "καλως", "μεταξυ", "ακριβως", "αναμεταξυ", "εντελως", "περα", "ευτυχως", "αντιπερα", "εξης", "ποτε", "ποσο", "καποτε", "καμποσο", "αλλοτε", "τοσο", "τοτε", "οσο", "τωρα", "οσοδηποτε", "ποτε", "πολυ", "οποτε", "πιο", "οποτεδηποτε", "λιγακι", "αμεσως", "σχεδον", "κιολας", "τουλαχιστον", "ηδη", "περιπου", "πια", "καθολου", "μολις", "διολου", "ακομη", "ολωσδιολου", "ακομα", "ολοτελα", "παλι", "μαλλον", "ξανα", "εξισου", "συνηθως", "προτυτερα", "μαλιστα", "νωρις", "ορισμενως", "χτες", "ισως", "χθες", "ταχα", "σημερα", "δηθεν", "αποψε", "πιθανον", "αυριο", "αραγε", "μεθαυριο", "Οχι", "οχι", "ναι", "περσι", "δεν", "προπερσι", "τοσο", "μη", "μην"

---

## **APPENDIX B - Data Cd**

---

## **APPENDIX C – Published Papers**

(In descending chronological order)

- Georgios Mamakis, Athanasios G. Malamos, J. Andrew Ware<sup>3</sup>, Ioanna Karelli, “Document Classification in Summarization”, Accepted for publication, Journal of Information and computing Science
- G. Mamakis, A. Malamos, J. A. Ware, “An Alternative Approach for Statistical Single-label Document Classification of Newspaper Articles”, Journal of Information Science, Sage Publications June, 2011 vol. 37 no. 3 293-303
- Athanasios G. Malamos, Georgios Mamakis and J. Andrew Ware, “Applying Statistic-based Algorithms for Automated Content Summarization in Greek language”, TEMU (Telecommunications and Multimedia) 2006, Heraklion Crete, 5-7 July 2006
- Mamakis G., Malamos A.G., Kaliakatsos Y., Axaridou A., Ware A., “An algorithm for automatic content summarization in modern greek language”, in Proceedings of IEEE-ICICT(International Conference on Information and Computer Technology) '05, ISBN 0-7803-9270-1, Cairo, Egypt, pp 579-591, 5-6 December 2005

## Document Classification in Summarization

Georgios Mamakis<sup>1</sup>, Athanasios G. Malamos<sup>2</sup>, J. Andrew Ware<sup>3</sup>, Ioanna Karelli<sup>4</sup>

<sup>1</sup>Faculty of Advanced Technology, University of Glamorgan, Trefforest, Wales and Department of Applied Informatics and Multimedia. Technological Educational Institute of Crete, Heraklion Crete, Greece Email: gmamakis@epp.teicrete.gr

<sup>2</sup>Department of Applied Informatics and Multimedia. Technological Educational Institute of Crete, Heraklion Crete, Greece, Email: amalamos@epp.teicrete.gr

<sup>3</sup>Faculty of Advanced Technology, University of Glamorgan, Trefforest, Wales, Email: jaware@glam.ac.uk

<sup>4</sup>Faculty of Philology, University of Crete, Rethymnon, Crete, Email: iwannakarelli@gmail.com

*(Received xxx 2005, accepted xxx 2005, will be set by the editor)*

**Abstract.** Document classification and document summarization have a fairly indirect relation as document classification fall into classification problems as opposed to document summarization, where it is treated as a problem of semantics. A major part of the summarization process is the identification of the topic or topics that are discussed in a random document. With that in mind, we try to discover whether document classification can assist in supervised document summarization. Our approach considers a set of classes, in which a document may be classified in, and a novel summarization scheme adapted to extract summaries according the results of the classification. The system is evaluated against a number of supervised and unsupervised approaches and yields significant results.

**Keywords:** Document classification, supervised document summarization, statistics

### 1. Introduction (Use “Header 1” Style)

One of the major areas of data engineering both nowadays and in the past is text management. Important work has been undertaken in the area since early in the history of Information Technology. Text management includes subjects as document classification and document summarization. Document classification refers to the automatic assignment of a random document to one (single-label) or more (multi-label) classes. Applications of document classification include spam mail recognition and decision support systems. Document summarization, on the other hand, refers to the extraction or generation of text from one or multiple sources, in a shortened form compared to the original source(s). In this paper, we examine whether the use of document classification can result in better summaries, or if it can yield significant results. Our motivation came from remarks regarding document summarization. The main motivation came from a generic summarization procedure template that was first proposed by Lin and Hovy in [1]. The authors proposed that one of the important factors in document summarization is the identification of the topics that are present in a document. Identifying the topics discussed in a document enables to some extent the identification of the important words that will assist in the final extraction or generation of the summary. In addition to that, Moens et al. [2] undertook research utilizing a

classification scheme to decide whether a random word in a document is a topic word (term) or not. Moreover, research by Barzilay et al. [3] tried to investigate news articles in conjunction to the topic they describe. However, their scope was to exploit characteristics that were domain-dependent, e.g. the pattern of authoring behind earthquake articles (location, size, victims). These research approaches led us to consider whether classification is an appropriate assisting tool in summarization tasks, not only in deciding if a random word is a topic word or a random sentence is a potential summary sentence as implied by [4] and [5], but rather in applying an adaptive approach on word importance, based on the class a random document may belong to. Therefore, instead of searching for the extraction of terminology that would result in identifying the potential topics of a document, we consider a set of class thesauri consisted of what we have automatically identified as terminology in the classifier training phase, classify the random document according to the lexicon that it adapts best to, and use this reference lexicon in extracting the most important sentences of the document as a summary.

The rest of the paper is organized as follows: In section 2, we provide background information on text classification, and insight on previous work we have undertaken in the area. In section 3, we present several document summarization approaches, and categorize them according to the scope and approach used, while in section 4, we validate theoretically our approach in supervised document summarization using classification, and analyze the main concepts behind our algorithms. In section 5, we provide an extended description of the limitations and algorithms that apply to our approach, while in section 6 we proved experimental results comparing several supervised and unsupervised algorithms. The final section of this paper concludes with future work in the area, underlining the feasibility of our approach.

## 2. Document Classification

One of the major problems in Machine Learning (ML) is deciding on the labeling of random input text into categories. Text classification or categorization has been an intriguing task, given that such decisions may not be always obvious. In order to tackle such problems, a number of approaches have been proposed such as statistical approaches, vector space models, artificial intelligence, decision trees and rules-based methods. Statistical classifiers are the most widely used generation of classifiers, since they are very efficient, very easy to construct and perform extremely quickly. Statistical classifiers include, among others, classifiers such as Naïve Bayes Classifier (NBC), Language Models and regression algorithms. Each algorithm provides a different approach in extracting the class of random input data, according to the number of labels it can assign. Thus, a second distinction, apart from the technology utilized, in document classification is referring to single-label classification, where the random case is assigned exactly one label, and multi-label classification where the classifier can assign random input to a set of potential classes.

A typical classifier consists of two discrete modules:

- A training phase, where the classifier is provided with a number of features and the class they correspond to, constructing a classification decision space
- An application phase, where the classifier decides on the class a random feature set approximates best, using the classification decision

Document classification is a special case of classification algorithms, where the input features are the document words and the output is a class or set of classes where a random document may belong to. However, generic classification approaches apply as well. The most commonly used statistical algorithm is NBC. NBC is a supervised statistical classification algorithm based on the Bayes theorem of statistical independence, assuming that each input feature value is statistical independent to any other input feature in the same feature set. Despite the naivety in such an approach, NBC has been proven to operate very

efficiently [6], outperforming more complex algorithms and approaches. Researchers that have modified and enhanced NBC in the past are [7,8,9]. However, it has been fairly recently suggested in [10], that NBC has a major drawback in its operation, that occurs when the set of training classes distribution is uneven. In such cases NBC behaves in a biased manner towards larger datasets. This has also been experimentally proven in our case as shown in [11].

Another commonly used statistical classification algorithm is Language Models (LMs). LMs are statistical models that instead of assuming statistical independence among features, use n-grams of features in both training and evaluation phase. The efficiency of LMs lies in the fact that they consider not only the existence of one word, but the co-existence of a sequence of words as e.g. San Francisco or Mona Lisa. Extensive work on LMs has been undertaken by [12] and [13]. It has been stated that unigram LMs approximate efficiently NBC results [14].

A common characteristic of both algorithms is that they are single-label classifiers. Multi-label classification is largely considered as an extension to single-label classification. Multi-label classification is generally achieved through a series of binary classifiers over multi-label training datasets to identify the classes a random document may belong to. Examples of research in the area includes modified kNN (k-Nearest Neighbors) approaches as the ones proposed by Cheng and Hullermeier in [15] and Zhang and Zhou in [16], or adaptations on algorithms such as SVM, proposed by Godbole and Sarawagi in [17].

### **3. Document Summarization**

Document Summarization refers to the process of extracting or generating shortened content from one or various sources. This content generally either answers to specific user questions or offers a more generic covering as many topics as possible. The size of the summary can be either proportional to the original source or absolute in number of words or sentences. Research in document summarization dates back in late 1950s, where Luhn's [18] and Baxendale's [19] work offered the basis upon which further research has been undertaken. Their pioneering work utilized statics and spatial document characteristics, as the means of evaluating the importance of sentences. Statistics-based algorithms engage term frequency in order to estimate the topics that may be discussed in a random document. On the other hand, location-based approaches exploit spatial document characteristics as sentence or paragraph location inside the document. The importance of each sentence or paragraph may be estimated through document analysis on the location of the main topic sentence, either through a template approach, by applying domain-specific templates of topic sentence occurrence, or through document monitoring and analysis over a series of uniform documents. As it has been suggested by Lin and Hovy in [1], summarization can be analyzed into three different tasks: a) identify the topics discussed in a document, b) evaluate the importance of each topic in the document, and c) create the summary either from extraction of the most important sentences as they appear in the document, or through generation of new sentences. Apart from extraction or generation, another categorization one can make in document summarization depends on the kind of information one requires from the original document. According to that categorization, we are referring to generic and question-based summarization. Generic summarization targets on providing a more general set of topics trying to cover as much information as possible from the original document. Question-based summarization, on the other hand, tries to provide information only on the topics desired by the user. A last categorization of document summaries results from the use of a training phase or external information that may be necessary for the system to operate. Thus, algorithms can be categorized in supervised or semi-supervised (use of a semantic lexicon or training set of documents) and unsupervised summarization (no external reference used).

Machine Learning, and especially document classification, has been used to assist document summarization in the past. Most approaches [4, 5] consider the problem of summarization as a binary classification problem, whether a sentence should be included in the summary or not. Additional work utilizes HMMs [20] to construct a feature formula that considers the possibility that a sentence is a summary sentence, given that its preceding one is included in a summary. Feature selection formulas have also been considered in approaches utilizing Neural Networks [21] and Genetic Algorithms [22], as well, where the systems are trained to combine in linear or log-linear functions the contribution of each feature from a feature set in the identification of summary sentences.

## **4. Our Approach**

### **4.1. Motivation**

As it may be obvious from the definition of both document classification and summarization, the most important part of the algorithm is the identification of the topics discussed in a document. Despite, the obvious differences in the outcome, both classification and summarization try to extract important information on what the potential topic hierarchy of the document may be. A direct outcome that motivated our interest was to research on the potentiality of document classification for the identification of the document subject. Our thoughts revolved around the semantic exploitation of words based on the importance according to the subject represented in a document. This implies the use of a set of pre-classified words or terms weighted according to their importance in every class. As it will be shown later on, research led to a classification and summarization approach that tried to refine the extraction of information according to the importance weight assigned per term per category. The expected outcome was a refined statistical summarization algorithm that would adapt term weights according to the category the document was estimated to belong.

### **4.2. Approach Overview**

The approach we considered resulted in a supervised summarization system. This required three discrete phases: a) classification training, b) classification, and c) summarization. Phases a and b have been extensively described in [11], while the summarization phase has been evolved by our initial thoughts presented in [23].

### **4.3. Overview of Classification**

Instead of utilizing NBC, LMs or any of the known statistical algorithms, we developed a supervised statistical classifier carefully adapted for assisting our summarization algorithm. The classifier adopts the idea of statistical independence proposed in NBC. In addition to that, we consider a normalized weight scheme, where each word contributes to a certain extent to all potential classes, according to term frequency and class size. The main difference with NBC or LM is the use of a summation instead of a product in evaluating the potential class of the document. Thus, the categorization approach is not very strict in assigning a document class. Moreover, since each term contributes unevenly to each potential class according to its occurrence, a finite class is only used as a reference for the selection of the appropriate term weight considering all potential class weights, rather than a define criterion for exclusive class weight. Thus engaging multi-label approaches was beyond our scope of research.

### **4.4. Nouns and Adjectives Importance**

Prior to a brief overview of the classification and summarization, it is vital to identify the elements that will be used in both classification and summarization. As it has been stated in [24] the topic information of a document is included mainly in nouns and noun-phrases, rather than in any other grammatical feature.



In our research, we extend this idea by isolating adjectives as well, since we consider them to denote descriptive information on the nouns of the random document – positive, negative or neutral. Yet, this information enhances topic identification as it enables a clearer distinction on the context of a term, given disambiguation of noun definitions. This approach is used in both classification and summarization. In order to successfully identify nouns and adjectives we have developed an algorithm extending work proposed by Porter [25] and adapted by Kalamboukis [26].

#### 4.5. The Classification Problem

Let  $i$  be a random noun, and  $D$  a random document featuring word  $i$ , and  $j$  a category the document may belong to. We are trying to compute what is the possibility for document  $D$  to belong to category  $j$  given that word  $i$  appears in  $D$ .

In order to compute this, we assign as  $w_{i,j}$  as the weight of word  $i$  in category  $j$  computed as

$$w_{i,j} = \frac{|tf(i,j)|}{\sum_{i=0}^n |tf(i,j)|} \quad (1)$$

where  $tf(i,j)$  the term frequency of word  $i$  in category  $j$ , and  $n$  the total number of unique words comprising category  $j$ .  $w_{i,j}$  in this context denotes the importance or contribution of noun  $i$  in category  $j$ . By dividing with the total number of the noun term frequencies of category  $j$ , we form a normalized weight factor for nouns in that category, in order to overcome NBC bias. This formula is influenced by the Term Frequency (TF) used in Information Retrieval algorithms (the first parameter of  $tf.idf$  algorithm). This approach also enables us to properly estimate the similarity of a random document to a category, since a great number of significant words of a category in a random document (high-weighted word observation), denotes greater similarity between the document and the category. The similarity factor of our approach is based on the probability of a random document  $D$  belonging in category  $j$ , if word  $i$  is present in document  $D$ . Given that word  $i$  is assigned a weight per class  $j$ , then this probability is calculated by

$$p_{i,j} = P(D \in j | i \in D) = \frac{w_{i,j}}{\sum_{j=0}^n w_{i,j}} \quad (2)$$

where  $n$  is the total number of categories the system can identify. This metric takes into account the cross-class importance of the words. This algorithm strongly resembles the Inverse Document Frequency metric used in Information Retrieval (the second parameter of the  $tf.idf$  algorithm).

The evaluation criterion that denotes a document belonging to a category is called similarity factor ( $sf$ ) and is calculated by

$$sf(D,j) = \frac{\sum_{i=0}^{Dwords} \frac{w_{i,j}}{\sum_{j=0}^m w_{i,j}}}{Dwords} \quad (3)$$

where  $m$  is the total number of categories available and  $D_{words}$  the word count of document  $D$ . Since  $sf$  is computed on the observed set of nouns extracted by the document, and not all words appear in category  $j$ , the contribution for each word present in the document is either 0 if the word is not present in category  $j$  or calculated according to (4). Dividing by  $D_{words}$  gives us the Expected Value for each category  $j$ . The system decides that the document belongs to the category which maximizes the similarity factor  $sf$

$$D \in j \Leftarrow \text{argmax}(sf(D, j)) \quad (4)$$

where  $\text{argmax}(sf(D, j))$  is the maximum similarity factor of Document  $D$  in categories  $j$ .

#### 4.6. Overview of Summarization

The summarization phase is based on the results acquired in the classification stage by considering the class the document was found to belong to. The summarizer uses the class lexicon to assign a proportional term weight as computed in (5). In order to extract the sentences that will be kept in the final summary, we compute the normalized sum of all term weights of a sentence. This is computed by

$$Score_{sent} = \sum_{sent} \frac{\sum p(i, j)}{|p(i, j)|} \quad (5)$$

The normalization on the average of term length is used as the bias elimination feature of larger sentences. Thus, only sentences consisted of more important words in a proportional manner are considered. The final length of the summary is manually decided, while the summary consists of the  $k$ -th highest scoring sentences in their appearance in the original document. We define this feature as term density. Other features used in summarization are term frequency, while positional characteristic has also been considered. The positional characteristic named relative paragraph-sentence position considers the relative position of a sentence in a paragraph, where the initial sentences are assigned a higher score. It is calculated using

$$PrScore_{sent} = \frac{(a - ((a-1) * k))}{paragraphsize - 1} * Score_{sent} \quad (6)$$

where  $a$  a prejudice score,  $k$  the  $k$ -th sentence and  $paragraphsize$  the number of sentences in the paragraph.

### 5. Methodology

The summarization methodology consists of four modules: a stemming module used to identify nouns and adjectives, a training step used to calculate term importance per class and form the class dictionary, a classification step which decides on the class a document may belong to, and a summarization module which extracts the topmost important sentences, according to the class the document was found to belong.

#### 5.1. Stemming module

Stemming is the process of identifying the unaltered part as it is conjugated (stem) from its suffix. Pioneering work in the area was undertaken by Porter [25], where he researched suffix stripping for the English language. His work provided the fundamentals for the respective work of Kalamoukis [26] for the Greek language. The importance of suffix string in Machine Learning has been underlined by Scott

and Matwin [27] as they state that it is almost always used in document classification and summarization.

Our work in stemming [23] is based on Kalamboukis work on Greek suffix stripping. In order to extract nouns and adjectives from a document, we grammatically engage a grammatically enhanced version of Kalamboukis stemmer. The stemmer not only identifies but also eliminates unimportant words through a number of stop list sets, comprised of common words such as articles, prepositions, pronouns and adverbs. The next step is to identify document verbs, through a set of common verb endings. A special case occurs between nouns, adjectives and passive voice participles in Greek language. While active voice participles are not conjugated and only interfere with certain name endings, passive voice participle can be conjugated in the same manner as nouns and adjectives, distinguishable only by a 3-character triplet ( $\mu\epsilon\nu$ ). Therefore a more extensive approach is engaged in order to acquire the final document noun set. The resulting data from this procedure is a stem table of important nouns and adjectives, on a sentence level. The stemmer's performance may be affected in cases where a noun can be used as an adverb (e.g.  $\alpha\lambda\eta\theta\epsilon\iota\alpha$ - meaning truth or really), however such drawbacks do not constitute a major problem on the system's efficiency.

## 5.2. Training Step

The training step is responsible for gathering and weighting class terminology. It is applied in the training phase of the algorithm once, and results in the set of weighted dictionaries per word, as acquired from the training documents. Each category algorithm is stemmed according to the previous algorithm and nouns are gathered and weighted according to formula (2) for each category. The resulting categories are used for reference both in the classification and the summarization modules of the algorithm. The approach has been presented extensively in [11]

## 5.3. Classification Step

The classification step is responsible for assigning a random input document to one of the available classes. Each document is compared to each of the available classes, and the most important category is decided according to average term weights for that category. Once the category is decided, its weighting scheme is used for the summary extraction. The algorithm has been extensively described in [11].

## 5.4. Summarization Step

The summarization step acquires word weights per sentence, extracts the topmost important sentences within the limits set, and rearranges the sentences in their original order. The summarization module is presented in the Fig. 1.

1. Let document  $D$ ,  $k$  percentage of summaries and class weight lexicon  $l_c$
2. Split  $D$  into array of sentences  $S_D$
3. For each  $s$  in  $S_D$
4. Let sentence weight  $s_w = 0$
5. For each word  $w$  in  $S_D$
6. If  $w$  belongs to  $l_c$
7.  $s_w = s_w + l_{cw}$
8.  $words = words + 1$
9. End If
10. End For
11. End For
12. Sort sentences descending according to  $s_w$
13. Keep top  $k$  sentences
14. Sort  $k$  sentences according to appearance in original document

Fig. 1. Summarization Module Algorithm

## 6. Evaluation of Results

A full system evaluation requires the estimation of both the classifier and the summarizer. The algorithms were tested against automatic evaluation metrics either through supervised or unsupervised approaches. More specifically, the evaluation of the classification is accomplished by manually classifying the documents in the training set, while for the evaluation of the summarizer we use ROUGE-1 [28] evaluation metrics system. The classifier was tested against the NBC implementation included in Mallet NLP toolkit [29], and 2 language model implementations taken from Lingpipe [30] NLP toolkit, denoted as LM-3 (trigram based) and LM-6 (sixgram based). The summarizer was tested against a sample summarizer based on the tf.idf metric, SweSum [31] adapted for Greek language, two baseline summarizers and Microsoft Summarizer package included in Microsoft Office 2007. Especially for our evaluation we developed a system that used tf.idf [32] as the weighting scheme on a noun and adjective level, as well as an Latent Semantic Analysis (LSA) approach, as it has been proposed by Gong and Liu [33], using the Singular Value Decomposition (SVD) implementation of JAMA [34] package.

### 6.1. Classification Results

We gathered a training and a test corpus from Greek online newspapers that were manually pre-classified into six distinct categories. The training corpus was made up of 1015 newspaper articles while the test corpus was comprised of 353 articles. The training and the test set were gathered during different years from various sources, in order to present as many different writing styles and vocabulary as possible. The six categories are Business and Finance, Culture, Health, Politics, Science and Technology, and Sports. The analysis according to our training scheme resulted in the following dictionary profiles per class depicted in Table 1.

Category	# of unique words	# of total words	Average word occurrence	Average weight
Business & Finance	5686	59777	10.51	0.000176
Culture	5750	26932	4.68	0.000174
Health	1181	3073	2.6	0.000847
Politics	7059	63218	8.96	0.000142
Science & Technology	3593	24429	6.8	0.000278
Sports	4696	15412	3.28	0.000213
Totals	27965	192841	6.139254653	0.000304909

Table 1. Class profiles

After applying classification on the test corpus, the results in Table 2 were acquired.

		Our Classifier	NBC	LM-6	LM-3
Correct	#	326	302	284	294
	%	92,35	85,55	80,45	83,29
Wrong	#	27	51	69	59

	%	7,65	14,45	19,55	16,71
Totals	#	353	353	353	353
	%	100	100	100	100

Table 2. Overall Classification Results

The results of Table 2 denote the overall efficiency of the algorithm in all categories given the class profiles of Table 1. More information on the general efficiency of the algorithm has been analyzed in [11] and goes beyond the scope of current research. As a brief conclusion we can see that the classification module significantly outperforms language models and NBC classifier, ensuring best performance for the summarization step.

## 6.2. Summarization Results

From the 353 articles comprising the test corpus we randomly selected 237 that were provided to a philologist for summary extraction. The philologist was unaware of our summarization extraction algorithm and was only generally guided to follow the same rules as the ones supplied to any of the summarization systems our algorithm was tested against. The philologist and the systems accordingly had to provide extracts of roughly 30% of the initial document with the sentences in order of original appearance. In addition to that she was also instructed to take notes on her summarization approach so as to both identify potential future enhancements to the system and explain the automatic results more adequately. Our summarization algorithm was tested against five other algorithms of supervised and unsupervised summarizers and their efficiency was automatically evaluated using ROUGE-1 metric [28]. The systems are a custom made algorithm using tf.idf (term frequency-inverse document frequency) metric as its weighting scheme, SweSum [31] summarization engine adapted for Greek language, Latent Semantic Analysis as proposed by Gong and Liu, a lead algorithm extracting the first 30% of the sentences of each random article (LEAD), a baseline algorithm extracting the first sentence of each article paragraph – a simplified approach over Baxendale’s [19] approach and Microsoft Summarizer. Since the last three algorithms are pretty straightforward, we will focus on SweSum, tf.idf and LSA, prior to commenting on ROUGE-1 evaluation metric and providing the results.

## 6.3. GreekSum – SweSum engine adapted for Greek language [31]

GreekSum is a summarization engine adapted from the SweSum summarization engine for the Greek language. GreekSum has been developed as a Master thesis by Pachantouris in KTH/Stockholm and is available online at <http://www.nada.kth.se/iplab/hlt/greeksum/index.htm>. Research initially concentrated on the Swedish language. To our knowledge, GreekSum is the only widely available Greek summarization engine. It produces extractive summaries of single documents by utilizing a series of features, as statistics, sentence position, document genre and keywords. It supports both supervised and unsupervised summarization. In our experiments, we used the supervised approach, since the author states it produces significantly better results than the unsupervised mode.

## 6.4. Tf.idf[32]

Trainable summarizers use a reference corpus for the estimation of word importance. This is accomplished using statistical analysis on term frequency. The problem trainable summarizers try to tackle is the extraction of topic terminology from a random document. Despite, shallow assumptions on the topic itself, since no semantic reference is made through simple statistics, the identification of important terminology, regardless of the topic inference, can approximate efficiently sentence importance.

Therefore, term frequency can be reduced to terminology extraction. However, terminology extraction on a single document can become very difficult considering the fact that highly frequent words may be generic words that do not represent any important meanings. Following this problem, tf.idf -an empirical approach- has been proposed that yields significant results.

Tf.idf has been an empirical metric extensively used in NLP as a decision making feature on word importance over a custom-built vocabulary set. It was first mentioned in [32], while extensive work has been undertaken, as e.g. in [33] and [34]. The main idea behind tf.idf is the elimination of commonly used words and identification of topic terminology. Tf.idf is generally calculated

$$tfidf_{word} = word / \sum word * \log(1 + train_{doc}) / (1 + train_{doc, word}) \quad (7)$$

where  $word$  denotes the occurrences of term  $word$  in a random document,  $\sum word$  the total number of words in the document,  $train_{doc}$  the total number of documents in the training set and  $train_{doc, word}$  the total number of documents in the training set having term  $word$ . As it may be obvious tf.idf promotes as terminology words that occur only in few documents in the training set as it results in a high idf, and secondarily those occurring many times in the random documents.

For the evaluation of our system we include a simplistic version of tf.idf, trained on the initial classification training corpus according to formula (7). Each sentence is scored against the average tf.idf score of its terms.

## 6.5. Latent Semantic Analysis

A commonly used NLP summarization method based on Singular Value Decomposition (SVD), patented in [37], is Latent Semantic Analysis (LSA). LSA tries to evaluate the contribution of a word in a text segment (extending from a sentence to document in cases of multi-document summarization) as well as the importance of a text segment featuring a word. LSA succeeds in both identifying noun phrases (San Francisco) and identifying the different topics presented in a document. The first step of the algorithm is to construct a matrix of terms by sentences. Considering that, generally, document terms ( $m$ ) are unequal to document sentences ( $n$ ),  $A$  is an  $m \times n$  matrix.  $A$  is a very sparse matrix as not all terms contribute to every sentence. Through SVD, matrix  $A$  is decomposed in

$$A = U \times \Sigma \times V^T \quad (8)$$

where  $U$  is a column-orthogonal matrix holding the left singular vectors,  $\Sigma$  is a diagonal matrix whose values are sorted in descending order and  $V$  is an orthonormal matrix holding the right singular vectors. As stated in (Gong & Liu, 2001), from a transformation point of view, SVD provides a mapping between each left singular vector (word) and each right singular vector (sentence). From a semantic point of view, SVD represents the analysis of the original document into concepts, captured into the singular vector space. In addition to that, it enables the establishment of strong relations between semantically related terms, as they will be projected very close to the singular vector space, as they share a great number of common words. The authors conclude that in SVD, each singular vector denotes a salient topic, whereas the magnitude of the vector denotes the importance of the topic. As stated by the authors of LSA [39], in contrast to identifying term co-occurrence, LSA tries to estimate the average meaning of a passage (e.g. sentence, paragraph or document) from the terms it consists of.

For the evaluation of our system, a system based on the fundamental work in LSA by [34] was developed as a test environment. The package used for the SVD was JAMA [35]. The algorithm proposed by the authors is based on the semantic representation of the SVD. More specifically, matrices  $\Sigma$  and  $V^T$  in SVD, are used to denote the value of the topics discussed in the document and the contribution of the each sentence in each topic respectively. Thus, Gong and Liu propose acquiring from the  $V^T$ , the sentence that has the highest singular value (column) for a given topic (singular index). Given that  $\Sigma$  is a diagonal matrix whose values are ordered descending, then it is safe to consider that the top  $r$  topics from  $\Sigma$  are represented by the top  $r$  columns from  $V^T$ . Therefore, the most significant topics are extracted, each one represented one sentence, thus reducing redundancy. In this research, LSA was considered as proposed by Gong and Liu, while motivated by [39], where the author stated that in small document sets stemming can improve performance, two LSA approaches were considered, one with and one without stemming.

## 6.6. ROUGE-N[28]

ROUGE-N is a recall oriented method proposed by Lin and Hovy. It has been used in various DUC conferences as an automatic evaluation metrics system. ROUGE-N treats word occurrences as n-grams, and tries to find the maximum number of n-grams between a candidate summary and a reference summary. ROUGE-N is calculated by

$$ROUGE - N = \frac{\sum_{S \in \{RefSum\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{RefSum\}} \sum_{gram_n \in S} Count(gram_n)} \quad (9)$$

ROUGE-N is not uniquely identified as a metric, but rather as a methodology of evaluation metrics, given its dependence on n-gram size. Moreover, the authors have proposed a series of variations to the initial algorithm as ROUGE-L, ROUGE-W and ROUGE-S. For our evaluation purposes, we utilized ROUGE-1 as it is considered to adapt better to the human understanding of summary.

## 6.7. Evaluation of the results

After comparing the summaries extracted by each of the algorithms with the human summaries using ROUGE-1, the results depicted in Table 3 where extracted:

Method	ROUGE-1
Our approach	0.4866
With tf.idf	0.5462
LEAD	0.517
GreekSum	0.5589
Microsoft Summarizer	0.4666
Baseline	0.4965
LSA with stemming	0.442
LSA without stemming	0.447

Table 3. Summarization Results

As it is depicted, the best performing algorithm is SweSum's adaptation for the Greek language, followed by tf.idf. This is not strange considering that both algorithms are trainable and precondition a

statistical or feature training analysis prior to the summarization process. The unexpected result was the performance of both LEAD and Baseline summarizers since they only use an overly simplified approach in extracting their sentences. Moreover, the performance of the summarization through classification seems to only outperform Microsoft Summarizer. After applying the algorithm with positional characteristics setting  $\alpha = 1.2$  in formula (6), the results acquired were:

Method	ROUGE-1
Our approach with positional characteristics	0.502
With tf.idf	0.5462
LEAD	0.517
GreekSum	0.5589
Microsoft Summarizer	0.4666
Baseline	0.4965
LSA with stemming	0.442
LSA without stemming	0.447

Table 4. Summarization Results with positional characteristics

As it can be seen, the inclusion of positional characteristics in our approach increases the efficiency of the algorithm about 3%. However, it still falls behind all knowledge-rich approaches and the LEAD approach. This can be explained if we consider how the human summarization process was carried out, which has also been validated Nenkova's [40] words regarding the efficiency of single-document summarization system's on newswire summarization tasks, where she states that performance is inefficient due to the structure of newspaper articles.

## 6.8. Human summarization approach

In order to fully understand the evaluation results, a number of findings have to be presented regarding the approach in the summarization tasks of the human summarizer. These have not been taken into consideration into any of the algorithms and are part of future work. As the philologist states: "Prior to presenting our findings, it is important to point out that these are indicative findings rather than objective results, since language cannot be sealed and qualitatively evaluated using objective measures. This happens due to both the liquidity of the language, which is considered as the Message, and other to parameters that play an important role in its examination; the author-transmitter and the reader-receiver of the Message. Thus, the results are presented without any notion of absoluteness, framed with theory proof when possible.

The first characteristic we identified and included in our summaries is the first sentence of a paragraph. It is usual both in journalistic and in essay writing, the context of the paragraph to be presented in the first sentence, the so-called thematic period.

The next element included in the summaries is the side heads that accompany the main document title. With regards to the latter I refer to the case where the text has side heads, while the paragraphs following them are omitted. In the case that the document has a numbered order, the element that we include in the summary is the first period that follows the numbered order, since this is the sentence that evaluates the meaning of those described before.

Apart from the elements included in the extract, equally important are the elements that were omitted from the summaries. Direct speech, for example, is an element that was omitted from the summaries in



most cases. In journalist articles, direct speech is used to explain plainly, using the people involved as roles, what has been described earlier by the author. Thus, it is safe to conclude that omitting direct speech from the summary automatically includes the period preceding direct speech.

A second element that was left out of the summaries is punctuation marks that introduce induction along with the text they include or presage. Examples are parentheses, dashes, and “e.g.”. These three elements further analyze the thought of the author and offer details that are not necessary for the summary.

An ambiguous issue came up in the management of interviews, the problem being the rapid change in the two people talking. For two reasons we included the questions of the journalists rather than the answers; first of all, the questions included the potential response of the answer, while also, the words of the interviewee have already been included either from the title or from the side head of the document."

## **6.9. Final Remarks**

Following these comments the good efficiency of the simplest algorithms as well as GreekSum (which uses partially spatial information in extracting sentences as a selection feature) can be explained. It also explains the performance of our approach, enabling, however, potential extensions using a series of features instead of only the classification. Moreover, another interesting feature will be a manual evaluation of the efficiency of the best scoring, the human summary and our algorithm's summary, since summary is usually of implicit nature and the subjective approach in what a human considers important or not.

## **7. Conclusions**

In this paper, we presented our thoughts on assisted document summarization through classification. The results have proven the potentiality of such an approach, however, they should also be validated against other algorithms using a common training set as the ones used in DUC conferences that would yield a better approximation on the algorithms efficiency. In addition to that, given the remarks supplied by the philologist regarding special requirements of newspaper articles, a potential extension would be either to include generic spatial characteristics in summary extraction or to extend the summarization algorithm with a knowledge-aware module that would automatically gather spatial characteristics on the probability of sentence selection according to position as it has been proposed in [19]. As a conclusion, document classification in summarization seems to be a feasible task, despite the limitations imposed by the extended training phase and the lack of pre-classified corpora and evaluation summaries.

## **8. Acknowledgements**

The authors would like to thank Dr. Dimitrios Karayannakis of the Technological Educational Institute of Crete for his help and guidance throughout this research with his remarks.

## **9. References**

- [1] C.Y. Lin., E. Hovy, "The Automated Acquisition of Topic Signatures for Text Summarization", Proc. of 18th conference on Computational linguistics, Vol. 1, pp.495-501, July - August 2000, doi:10.3115/990820.990892
- [2] M. F. Moens, R. Angheluta and J. Dumortier, "Generic technologies for single- and multi-document summarization", Journal of Information Processing and Management, Elsevier, vol 41, pp 569-586, 2005
- [3] R. Barzilay, L. Lee, "Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization", in Proc. HLT-NAACL, 2004
- [4] J. Kupiec, J. Pedersen and F. Chen. "A Trainable Document Summarizer ", Proceedings of the 18<sup>th</sup> International Conference on Research in Information Retrieval (SIGIR ' 95), pp. 55-60, 1995

- [5] C. Aone, M. Okunowski, J. Gorinsky, J., and B. Larsen, "A scalable summarization system using NLP", *Proc. of the ACL '97 EACL '97 Workshop on Intelligent Scalable Text Summarization*, pp. 66-73, 1997
- [6] I. Rish, "An empirical study of the Naive Bayes Classifier", in *Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, pp 41-46, 2001
- [7] A. K. McCallum and K. Nigam, "A comparison of Event Models for Naive Bayes Text Classification", in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 41-48, AAAI Press, 1998
- [8] K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM", in *Machine Learning*, Vol 23, Issue 2-3, Special Issue on Information Retrieval, pp 103-134, Kluwer Academic Publishers, ISSN:0885-6125
- [9] W. Dai, G. R. Xue, Q. Yang, Y. Yu, "Transferring Naive Bayes Classifiers for Text Classification", in *Proc. of the 22nd AAAI Conference on Artificial Intelligence*, pp540-545, AAAI Press, 2007
- [10] J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, "Tackling the poor assumptions of Naive Bayes Text Classifiers", in *Proc. of the 20th International Conference on Machine Learning*, ICML-2003
- [11] G. Mamakis, A.G. Malamos, J.A. Ware, "An alternative approach for statistical single-label document classification of newspaper articles", *Journal of Information Science*, Vol. 37 (3), pp 293-303, SAGE publications, June 2011
- [12] M. Srikanth, R. Srihari, "Bitern language models for document retrieval", in *Proc. of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, 425-426, Tampere, Finland, 2002
- [13] W. B. Croft, "Language models for Information Retrieval", in *Proc. of the 19th International Conference on Data Engineering*, Bangalore, India, 2003
- [14] F. Peng, D. Schuurmans, S. Wang, "Augmenting Naive Bayes Classifiers with Statistical Language Models", in *Information Retrieval*, Vol 7, Issues 3-4, pp 317-345, 2004, Kluwer Academic Publishers, ISSN:1573-7659
- [15] W. Cheng, E. Hullermeier, "Combining instance-based learning and logistic regression for multilabel classification", *Machine Learning* vol76, pp211-225, 2009, doi:10.1007/s10994-009-5127-5
- [16] Zhang, M.-L., & Zhou, Z.-H. "ML-kNN: A lazy learning approach to multilabel learning", *Pattern Recognition*, 40(7), pp 2038-2048, 2007
- [17] Godbole, S. & Sarawagi, S., "Discriminative Methods for Multi-labeled Classification", *Lecture Notes in Computer Science*, 2004, Vol. 3056/2004, pp 22-30, DOI: 10.1007/978-3-540-24775-3\_5
- [18] H.P. Luhn, "The Automatic Creation of Literature Abstracts", *IBM Journal*, pp 159-165, April 1958
- [19] P.B. Baxendale, "Machine-Made Index for Technical Literature – An experiment", *IBM Journal*, pp 354-361, October 1958
- [20] J. Conroy, and D. O'Leary, "Text summarization via hidden markov models and pivoted QR matrix decomposition", *24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR'01.*, 2001
- [21] K. Svore, L. Vanderwende, L., and C. Burges, "Enhancing Single-document Summarization by Combining RankNet and Third-party Sources", *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 448-457, Prague, Czech Republic, 2007
- [22] M. Fattah, and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization:", *Computer Speech and Language*, Vol 23, pp.126-144, 2009

- [23] Mamakis G., Malamos A.G., Kaliakatsos Y., Axaridou A., Ware A., "An algorithm for automatic content summarization in modern greek language", in Proc. of IEEE-ICICT '05, pp 579-591, ISBN 0-7803-9270-1, Cairo, Egypt, 2005
- [24] C. Bouras, V. Tsogkas, "Improving Text Summarization Using Noun Retrieval Techniques", in Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 5178, pp593-600, 2010
- [25] M. F. Porter, "An algorithm for suffix stripping", in *Program*, Vol14, No 3, pp 130-137, 1980
- [26] T. Z. Kalamboukis, "Suffix stripping with modern greek", in *Program: electronic library and information systems*, Vol. 29, No 3, pp313 - 321, 1995
- [27] S. Scott, S. Matwin , "Feature Engineering for Text Classification", in Proceedings of the Sixteenth International Conference on Machine Learning, (1999), 379-388, Morgan Kauffman Publishers, ISBN:1-55860-612-2
- [28] C. Y. Lin, and E. H. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics", Proc of 2003 LanguageTechnology Conference (HLT-NAACL 2003),2003
- [29] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit.", <http://mallet.cs.umass.edu> (last accessed 20/7/2010)
- [30] Alias-I, LingPipe 4.0.0. <http://alias-i.com/lingpipe> (last accessed 20/7/2010)
- [31] G. Pachantouris, "GreekSum – A Greek Text Summarizer.", Master Thesis, Department of Computer and Systems Sciences, KTH – Stockholm University, 2005
- [32] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval". *Journal of Documentation* 28 (1): pp 11-21, 1972, doi:10.1108/eb026526.
- [33] Y. Gong and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis", *Proc. SIGIR'01*, pp 19-25, September 2001
- [34] *JAMA: Java matrix package*. Retrieved 7 10, 2011, from JAMA: Java matrix package: <http://math.nist.gov/javanumerics/jama>
- [35] H.C. Wu, R.W.P. Luk, K.F. Wong, K.L. Kwok. "Interpreting TF-IDF term weights as making relevance decisions". *ACM Transactions on Information Systems*, Vol 26, No 3, pp 1–37, 2008 doi:10.1145/1361684.1361686
- [36] G. Erkan, D. R. Radev, "LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization", *Journal of Artificial Intelligence Research*, Vol 22, pp 457-479, 2004
- [37] S. Deerwester, S. Dumais, G. Furnas, R. Harshman, T. Landauer, K.. *Patent Computer information retrieval using latent semantic structure*. USA, 1988
- [38] Landauer, T., & Dumais, S. (n.d.). *Latent Semantic Analysis*. Retrieved 6/6/ 2011, from Scholarpedia: [http://www.scholarpedia.org/article/Latent\\_semantic\\_analysis](http://www.scholarpedia.org/article/Latent_semantic_analysis)
- [39] S. Dumais, "Latent Semantic Analysis" , *Annual Review of Information Science and Technology* , Vol. 38 (1), pp. 188-230, 2004
- [40] A. Nenkova, "Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference", *20th National Conference on Artificial Intelligence (AAAI 2005)*. Pittsburg.

# **Applying Statistic-based Algorithms for Automated Content Summarization in Greek language**

Athanasios G. Malamos<sup>1</sup>, Georgios Mamakis<sup>2</sup> and J. Andrew Ware<sup>3</sup>

<sup>1,2</sup> Department of Applied Informatics & Multimedia, Technological Educational Institute of Crete, {amalamos, gmamakis}@epp.teicrete.gr

<sup>3</sup> School of Computing, University of Glamorgan, jaware@glam.ac.uk

## **ABSTRACT**

Summary extraction for generic unstructured documents has been a major research issue since the 'birth' of Informatics. Early research on automated summary extraction and content manipulation dates back to the middle of the previous century, when the initial content manipulation algorithms were developed. Since then, major advances have been made, utilizing semantics and artificial intelligence techniques to help achieve a better level of accuracy, both in the contents and in the speed of extraction of the excerpts. Under this scope we have developed a series of algorithms, primarily targeting on extracting summaries for Greek language documents, based on statistical and semantic approaches. In this paper we present an application of these algorithms and results of accuracy as opposed to human extracted summaries. This work is supplementary to initial research undertaken by authors (Malamos et al., 2005).

## **KEYWORDS**

automatic document classification - summarization, statistical algorithms

## **1. Introduction**

A common factor in modern research area is the content handling efficiency each application is capable of providing to the end user. A high level of content handling efficiency implies a user-friendly, automated environment with low computational requirements while at the same time the summary will be close enough, if not identical, to the one produced by a qualified human. In order to be able to develop such a system it is necessary to identify the necessary steps, regarding content handling. Common approaches include stemming (extracting a form of a word that remains unchanged regardless of the actual use of the word in a sentence), domain creation (creating sets of words that are associated with one another in terms of meaning or common occurrence), document classification (identifying the subject of the document), document summarization (producing either by extraction or by generation a small summary of a document). The first research paper that introduced automatic document summarization as one of the active research areas in informatics was presented by Luhn in (Luhn, 1958), where the author provides a statistic, keyword-based approach to extracting summaries. This work makes Luhn the pioneer in the initiation of semantics in computer processing. Since then have been introduced several approaches for summary extraction or generation other utilizing statistics, as in (Luhn, 1958) and (Edmundson, 1969) or syntactical analysis, as in (Baxendale, 1958), (Saggion and Lapalme, 2000), (Teufel and Moens, 1997), (Ono et. al., 94) and (Barzilay and Elhaddad, 1997). Later research outcomes involve also artificial intelligence techniques like Neural Networks (Kaikhah, 2004) and genetic algorithms (Jaoua and Ben Hamadou, 2003). In this paper, we extend automatic content summarization algorithms proposed by authors in (Malamos et al., 2005). These algorithms utilize statistical techniques and grammatical features of Greek language in order to

efficiently extract summaries of Greek documents. In the algorithm we present in this paper we enhance our system with the ability to handle unidentified grammatical elements like “active voice participles”. This improvement leads to a more qualitative summary and improves the overall efficiency. The rest of this paper is structured as follows: In section 2 we present the innovation of our system as opposed to other approaches, in section 3 we present algorithms for the development of an automatic summarization system, in section 4 we present an example of the operation of the system on a newspaper article, while in section 5 we present future work needed in order to enhance the efficiency of our system.

## **2. Motivation**

Our scope is to develop a fully automated document summarization system for Greek language articles. The improvement of the system we present to corresponding systems is that we utilize nouns instead of keywords both in classification and in summarization steps. Thus, our approach is advantageous over keyword-based systems as it does not require any preloading of keywords, and therefore the system can be fully-automated and easily adaptable and applicable to any thematic subject. This is accomplished through an enhanced stemming algorithm that can not only acquire stems of words but can also identify nouns in a sentence or document. This algorithm is also used to form thematic domains through accumulating the nouns of pre-classified documents and comparing the extracted set of noun stems per domain and filtering out the common ones. Each document is initially classified and the summary is finally extracted through a real-time statistical analysis of the nouns identified in the document as opposed to the thematic domains.

## **3. Methodology**

In order to state a summarization algorithm, one should be able to identify grammatical and syntactical requirements of the target-language. These requirements stem from grammatical and syntactical rules that define the structure and word form of each element belonging to the language’s vocabulary. Variations of word forms in a language implies that the initial step of the algorithm is to isolate the word part (or stem) that remains unchanged in any given grammatical form of that word, in order to identify the word itself in a document. Secondly, it is vital to assess the contribution of any given word inside the document to be summarized. We have identified that in Greek language (Malamos et. al, 2005), for example, nouns are essential in extracting the content of a document. Therefore, our stemming algorithm should be capable of not only extracting stems of words, but isolating the nouns of a document, as well. In order to produce a summary of documents, it is necessary to classify them to thematic domains. This approach resembles the one proposed by (Leskovec et. al. 2004), and is utilized to avoid ambiguity between words. Thus, the second step of our algorithm is a semantic classification of target-document. Classification is based on presence of nouns that belong to a specific domain in relation to the total number of nouns that exist in the document. Since a document is classified in a domain, the last step of the algorithm is to extract the summary. The system filters out the document in a by-sentence manner. The algorithm utilizes  $hr$  factor that is statistically related to the number of nouns in a sentence that belong to the domain as opposed to the entire set of nouns in the sentence. Apart from these stages another preparatory stage is required for the creation of the domains. In order to extract the set of words forming a domain we have used a corpus of 1080 Greek articles taken from newspapers. Then, using the

stemming algorithm, we isolated the nouns of each document and by considering their occurrences we formed a set of fifteen (15) domains.

### 3.1 Step 1 - Stemming Algorithm

As stated before, we have identified that nouns are important word elements in the extraction of a summary. The stemming algorithm we have developed is targeted on extracting stems of nouns regardless of their clause and disregarding other word elements, and is based on portions of Porter's stemming algorithm (Porter, 1980) and the Greek stemming algorithm proposed in (Kalaboukis, 1995). Greek noun endings are shown in Table1. Still noun endings are not enough in extracting a noun as active voice participles (typically ending in -ώντας or -οντας), passive voice participles (typically ending in -μένος, -μένη, -μένο for masculine, feminine and neutral genders in nominative case, respectively) and adjectives (sharing exactly the same endings with nouns) may interfere with the isolation of a noun. Our stemming algorithm can discriminate between nouns and active voice participles and passive voice participles, but fails when referring to adjectives. Still since adjectives are not domain specific words (may be used in any domain), an efficient corpus of documents for the creation of the domains will disregard them.

Noun genders	Possible Noun Endings
Masculine	ας, α, αδες, αδων, ες, ων, ης, η, ηδες, ηδων, ες, εδες, εδων, ους, ουδες, ουδων, ος, ου, ο, ε, οι
Feminine	α, ας, ων, ες, αδες, αδων, η, ης, ες, εις, εων, ω, ως, ος, ου, ο, οι, ων, ους
Neutral	ο, ου, α, ων, ατα, ατων, ι, ιου, ια, ιων, υ, ιου, ος, ους, η, , -α, ατος, ας, ως, ωτος, των

Table 1. Nouns endings

The same algorithm which is shown later on is used both to create the semantic domains and to isolate the nouns of a document. The algorithm is as follows:

#### *Pre-processing stage*

*Create stop-list  $S_j$  of noun endings for masculine, feminine and neutral nouns*  
*Create stop-list  $AP_k$  of nouns resembling active voice participles (mostly names)*

#### *Stage 1 Create the dictionary and exclude active voice participles*

*Let  $D$  a document,  $w$  the words of a document,  $e\}$  an array containing the last letters of the word  $w$  and  $i$  the number of letters taken from a word,  $dict$  the dictionary of nouns and  $pe$  active voice participle endings.*

#### *Begin*

*For each  $w$  in  $D$*

*If  $e\{5\} \in pe$  and  $w \notin AP_k$  then "assume the last five letters of the word"*

*Disregard the word*

```

Else if  $e\{5\} \in pe$  and  $w \in AP_k$  then
Add  $w-e\{2\}$  to dict          "remove the last two letters of the word to acquire its stem"
Else
For  $i=1$  to 4
If  $S_j = e\{1...i\}$  then          "assume the last  $i$  letters of the word"
Add  $w-e\{1...i\}$  to dict        "add the stem of the word in the dictionary"
Else
Disregard the word
End if
End for
End if
End for
End

```

*Stage 2 Calibrate the dictionary to avoid participles*

```

Begin
For each  $w$  in dict
If  $e\{3\} = \langle \mu\acute{\epsilon}\nu \rangle$  or  $\langle \mu\epsilon\nu \rangle$  then      "take the last 3 letters of the stemmed words"
Remove  $w$  from dict
Endif
Endfor
End

```

### 3.2 Off-line Step – Domain Creation

Our domain creation algorithm takes advantage of the occurrence of a noun in a document of the corpus. If a noun appears in more than four of the 15 domains then it is disregarded as insignificant. The corresponding algorithm follows:

*Let  $\{D_i\}$  the set of domains,  $Stem_i$  the  $i$ -th stem of a word in a domain,  $occ(Stem_i, \{D_i\})$  the number of occurrences of  $Stem_i$  in different domains,  $occ(Stem_i, D_i)$  the number of occurrences of  $Stem_i$  in a domain and  $d_i$  the documents forming a domain*

*Stage 1 Accumulate the words*

```

For each  $d_i$  in  $D_i$ 
For each  $Stem_i$  in  $d_i$ 
If  $Stem_i \notin D_i$  then
Add  $Stem_i$  to  $D_i$ 
Else
 $occ(Stem_i, D_i) = occ(Stem_i, D_i) + 1$ 
Endif
Endfor
Endfor

```

*Stage 2 Domain tuning*

```

For each  $Stem_i$  in  $D_i$ 
If  $Stem_i \in D_i$  Then
If  $occ(Stem_i, \{D_i\}) < 5$  then
 $occ(Stem_i, \{D_i\}) = occ(Stem_i, \{D_i\}) + 1$ 
else
Delete  $Stem_i$  from every  $D_i$ 
endif
endif
endif
endfor

```

### 3.3 Step 2 - Document Classification algorithm

In order to effectively extract a summary of an article, it is vital to identify the thematic subject of the document, in order to verify the semantics of the words used and calibrate their use. We adopt a statistic approach to verify the domain the document at hand. A document belongs to a domain by calculating  $d = (V \cap D) / (V)$ , where  $d$  is the domain factor of a document,  $V$  the nouns in the document and  $D$  the nouns constituting the domain. If  $d > 30\%$  then the document is considered to belong to domain  $D$ . The limit is set arbitrarily. Using this formula, a document may be found to belong to more than one domain, resembling our approach in the set nouns forming a domain. The initial step used in this algorithm is, as in domain creation step, our stemming algorithm. The resulting noun stems constitute the set of words compared to the noun stems of the domain. The corresponding algorithm is:

*Let*  
*BD {D1..Dr} set of domains that the doc belongs to*  
*Di{1..n} set of nouns of a i-th semantic domain*  
*V{1..k} set of nouns of Doc*

**"Step 1. Text Classification"**  
*Begin*  
*For i=1 to n*  
 $d(i) = (|V \cap Di|) / |V|$   
*if  $d(i) > 0.1$  then*  
 $BD = BD \cup \{i\}$  *"Doc belongs to Di semantic domain"*  
*end\_if*  
*end\_for*  
*End*

### 3.4 Step 3 – Document extraction

The final step of the algorithm, extracts the summary of the document, in a statistic and domain-oriented approach. Factor  $hr$  is computed as:

$$hr = f \cap D / f$$

where  $f$  represents the set of noun stems of a sentence of the document and  $D$  the set of noun stems constituting the domain. A sentence is considered to be any sequence of words between the beginning of a paragraph and a dot, between two dots, between a dot and a question-mark and vice versa, between a semicolon and a dot and vice versa and between an exclamation mark and a dot and vice versa.  $hr$  factor along with the absolute position of a sentence in the document constitute a hash table. The summary is extracted by constructing a document with the sentences having the greater  $hr$  factors in their relative position to one another. The number of sentences to be included is decided by the number of words, the user wants to include in the resulting summary. The corresponding algorithm is:

*Doc: the document to summarize*  
*MScript{1..p} set of sentences of the document ( $Doc = MScript_1 \cup MScript_2 \cup \dots \cup MScript_p$ )*  
*|MScript<sub>p</sub>| is the number of words (all kind of words, verbs, nouns, adjectives....) of the p-th sentence*  
*HTable(SentenceIndex, HitRatio) Hash table including the sentence (Mscript) SentenceIndex  $\in \{1..p\}$  and the HitRatio of the sentence*  
*BD {D1..Dr} set of domains that the doc belongs to*  
*Di{1..n} set of nouns of a i-th semantic domain*  
*F<sub>j</sub>{1..m} set of nouns of the j-th sentence MScript<sub>j</sub> in the document ( $V = F_1 \cup F_2 \cup \dots \cup F_p$ )*  
*a the number of words of the summary we intent to produce*



*Summary(j) a table of the sentences that will produce the summary*

*Begin*

*For each  $D_i \in BD$*

*For  $j=1$  to  $p$       “ $p$  is the number of sentences”*

*$hrj = hrj + f_j \cap D_i / f_j$*

*HTable( $j, hrj$ )*

*end\_for*

*end\_for each*

*Sort HTable ASC hrj “Sort HTable in Ascending order of hrj”*

*HTable\_index=1*

*While (no\_of\_words\_in\_summary <  $a$ ) and (HTable\_index <=  $p$ )*

*$j = \text{HTable}(\text{HTable\_Index}, 1)$  “content of HTable in row HTable\_Index and column 1”*

*Summary( $j$ ) = MScript $_j$*

*no\_of\_words\_in\_summary = no\_of\_words\_in\_summary + |MScript $_j$ |*

*HTable\_Index = HTable\_Index + 1*

*End\_While*

*For  $j=1$  to  $p$*

*Final\_Summary = Final\_Summary + Summary( $j$ )*

*End\_for*

*End*

#### 4. Application example

We have developed an application based on the algorithms presented on preceding chapters, in order to extract automatic summaries of newspaper articles identified to belong to one of the subjects recognized by our system. The application was initially developed to help evaluate the efficiency of the aforementioned algorithms, when compared to human extracted summaries. We will provide a typical example of machine-made categorization and summary on a newspaper article in Greek and its English translation.

Έντονη συναλλακτική δραστηριότητα σε τραπεζικούς τίτλους της Σοφοκλέους καταγράφηκε τις τελευταίες ώρες της συνεδρίασης της Τετάρτης, γεγονός που επιβεβαιώνει το μεγάλο ενδιαφέρον των θεσμικών επενδυτών για τις ελληνικές τράπεζες. Μεγάλο πακέτο μετοχών της Εθνικής Τράπεζα της Ελλάδος άλλαξε χέρια σήμερα. Πρόκειται για 2037644 τεμάχια συνολικής αξίας 77.83 εκατ. ευρώ. Η πράξη την οποία εκτέλεσε η Εθνική Χρηματιστηριακή, έγινε στα 38,20 ευρώ. Με βάση έγκυρες πληροφορίες οι μετοχές ανήκαν στον οίκο Fidelity, και αγοράστηκαν στην παραπάνω τιμή από την Citigroup προκειμένου η τελευταία να τις κάνει placement σε πελάτες της. Πολλά ήταν τα πακέτα που έγιναν σε μετοχές της τράπεζας Κύπρου, μεταξύ 5,70 έως 5,74 ευρώ. Συνολικά 2.640.000 μετοχές άλλαξαν χέρια. Παράλληλα όμως πακέτα, σαφώς μικρότερα με πωλητές ξένους και αποδέκτες ξένους διενεργήθηκαν και σε πολλές ακόμη τράπεζες. Στην Τράπεζα Πειραιώς άλλαξε χέρια πακέτο 150.000

Intense transactional processing in bank capitals in Sophokleous was recorded during the last hours of the Wednesday's session, assuring the great interest of statutory investors in the Greek banks. A great proportion of the Hellenic National Bank was transferred today. We refer to 2037644 stocks costing about 77.83 m €. The transaction was made by Ethniki Chrimatistiriaki and cost about 38.20€ per stock. According to valid sources the stocks belonged to Fidelity trust, and were bought by Citigroup in order to be placed their customers. A lot of packages were Cyprus Bank stocks, valuing between 5.70 € and 5.74€. A total of 2,640,000 were transacted. However smaller packages with both buyers and sellers coming from abroad were made in a number of banks. In Peiraios bank 150,000 stocks were transacted in a package at 19.80€ per stock, costing a total of 2,97 m € and in Alpha Bank 144.504 stock were transacted at 29.14€, costing up to 4,21 m €. This financial movement, according to specialists, implies that no massive immediate profit liquidation from great statutory investors, which are the

<p>μετοχών στα 19,80 ευρώ συνολικής αξίας 2,97 εκατ. ευρώ, στην Alpha Bank άλλαξε χέρια πακέτο 144.504 μετοχών στα 29,14 ευρώ συνολικής αξίας 4,21 εκατ. ευρώ. Η κινητικότητα αυτή δείχνει, σύμφωνα με εκτιμήσεις παραγόντων της αγοράς, πως δύσκολα θα δούμε, άμεσα τουλάχιστον, αθρόες ρευστοποιήσεις κερδών από μεγάλους ξένους θεσμικούς επενδυτές, που αποτελούν και τους πιο ενεργούς επενδυτές στην παρούσα φάση στη Σοφοκλέους.</p> <p>Taken from <a href="http://www.imerisia.gr">www.imerisia.gr</a> (25/1/2006)</p>	<p>fundamental and most active investors currently in Sophocleous, will take place.</p>
--	---

**Table 2. Document to be summarized**

The classification process according to the algorithm described above produced the following results for the specific article:

*Politics: 0.22, Sports: 0.15, Business & Finance: 0.28, International: 0.22*

The time needed to extract the summary was approximately 38 seconds. In order to extract the summary, we selected the highest scoring domain, setting a threshold  $t=0.3$ , implying that the extracted summary will be around 30% of the original document. The extracted summary is depicted in Table 3.

<p>Έντονη συναλλακτική δραστηριότητα σε τραπεζικούς τίτλους της Σοφοκλέους καταγράφηκε τις τελευταίες ώρες της συνεδρίασης της Τετάρτης, γεγονός που επιβεβαιώνει το μεγάλο ενδιαφέρον των θεσμικών επενδυτών για τις ελληνικές τράπεζες.</p> <p>Πρόκειται για 2037644 τεμάχια συνολικής αξίας 77.83 εκατ. Ευρώ. Με βάση έγκυρες πληροφορίες οι μετοχές ανήκαν στον οίκο Fidelity, και αγοράστηκαν στην παραπάνω τιμή από την Citigroup προκειμένου η τελευταία να τις κάνει placement σε πελάτες της.</p>	<p>Intense transactional processing in bank capitals in Sophocleous was recorded during the last hours of the Wednesday's session, assuring the great interest of statutory investors in the Greek banks We refer to 2037644 stocks costing about 77.83 m €. According to valid sources the stocks belonged to Fidelity trust, and were bought by Citigroup in order to be placed their customers.</p>
--	--

**Table 3. Summarization with  $t=0.3$**

The extracted summary was created at about 28 seconds.

Setting a lower threshold would result in a smaller extracted summary. The extracted summary in the case where the threshold was  $t=0.1$  (10% of the original document) is depicted on Table 4.

<p>Έντονη συναλλακτική δραστηριότητα σε τραπεζικούς τίτλους της Σοφοκλέους καταγράφηκε τις τελευταίες ώρες της συνεδρίασης της Τετάρτης, γεγονός που επιβεβαιώνει το μεγάλο ενδιαφέρον των θεσμικών επενδυτών για τις ελληνικές τράπεζες.</p>	<p>Intense transactional processing in bank capitals in Sophocleous was recorded during the last hours of the Wednesday's session, assuring the great interest of statutory investors in the Greek banks</p>
---	--

**Table 4. Summarization with  $t=0.1$**

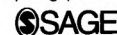
## 5. General conclusions and future work

In this paper we presented a system capable of producing summaries of Greek newspaper articles, based on statistical methods. Currently the system is undergoing an initial evaluation that will produce necessary results that would verify the enhancements needed to be made. The evaluation scheme followed is based on human extracted summaries and cross-examination of the results of the machine made summaries. The summaries are checked for efficiency, understanding and cohesion. The next step would include the application of known semi-human evaluation processes as  $tf*idf$  (term frequency and inverse document frequency) that would assist the recognition of potential disadvantages of both the algorithms and the corresponding system, as well as the fine-tuning required to the thresholds set for the operation of the algorithm. In our future plans, we consider the encapsulation of artificial intelligence techniques (as neural networks and genetic algorithms), in order to accelerate the extraction of results, while enhancements to the statistics algorithms are constantly made.

## 6. References

- Barzilay R, Elhadad. M. ,“Using Lexical Chains for Text Summarization”, in *Mani, I., and Maybury, M., eds., Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 11 July 1997, pp. 10-17.
- Baxendale P.B., “Man-made index for technical literature: an experiment”, *IBM Journal of Research and Development*, 2, 4, 1958, pp. 354-361.
- Edmundson H.P., “New methods in automatic abstracting”, *Journal of the Association for Computing Machinery*, 16, 2, 1969, pp. 264-285.
- Jaoua M., Ben Hamadou A., “Automatic Text Summarization of Scientific Articles Based on Classification of Extract's Population”, 4th International Conference of Computational Linguistics and Intelligent Text Processing, CICLing 2003 Mexico City, Mexico, February 16-22, 2003, , Lecture Notes in Computer Science Volume 2588 / 2003, pp. 623 – 634
- Kaikhah K., Automatic text summarization with neural networks *In Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, 22-24 June 2004 pp. 40- 44 Vol.1
- Kalamoukis TZ, “Suffix stripping with modern Greek”, *Program*, 29, 3 (1995), pp 313-321
- Luhn H.P., “The automatic creation of literature abstracts”, *IBM Journal of Research and Development*, 2, 4, 1958, pp. 159-165
- Malamos A.G., Mamakis G, Kaliakatsos Y., Axaridou A., Ware J.A., “An Algorithm for Automatic Content Summarization in Modern Greek Language”, *In proceedings of, ITI 3rd International Conference on Information & Communications Technology (ICICT 2005)*, Cairo, Egypt, 5-6 December 2005
- Ono K., Sumita K., Miike S., “Abstract generation based on rhetorical structure extraction”, *In the Proceedings of the 15th International Conference on Computational Linguistics - COLING'94*,. Kyoto, Japan (1994) Vol. 1, pp. 344-348
- Porter. M.F., “An Algorithm for Suffix Stripping”, *Program*, 14, 3, July 1980, pp. 130-137.
- Leskovec J., Grobelnik M., Milic-Frayling N., “Learning semantic graph mapping for document summarization”, *Jozef Stefan Institute, Slovenia*, June 2004.
- Saggion H., Lapalme G., “Concept Identification and Presentation in the Context of Technical Text Summarization”, *Automatic Summarization Workshop at NAACL/ANLP'2000*, Seattle, WA, pp 1-10
- Teufel S., Moens M., “Sentence extraction as a classification task”, in *Mani I., and Maybury, M. eds.*, 1997, pp. 58-65.

# An alternative approach for statistical single-label document classification of newspaper articles

Journal of Information Science  
1-11  
© The Author(s) 2011  
Reprints and permission: [sagepub.co.uk/journalsPermissions.nav](http://sagepub.co.uk/journalsPermissions.nav)  
DOI: 10.1177/0165551511403543  
[jis.sagepub.com](http://jis.sagepub.com)  


**Georgios Mamakis**

Technological Educational Institute of Crete, University of Glamorgan, Wales, UK

**Athanasios G. Malamos**

Technological Educational Institute of Crete, Greece

**J. Andrew Ware**

Department of Computing and Mathematical Sciences, University of Glamorgan, Wales, UK

## Abstract

Text classification is one of the most important sectors of machine learning theory. It enables a series of tasks among which are email spam filtering and context identification. Classification theory proposes a number of different techniques based on different technologies and tools. Classification systems are typically distinguished into single-label categorization and multi-label categorization systems, according to the number of categories they assign to each of the classified documents. In this paper, we present work undertaken in the area of single-label classification which resulted in a statistical classifier, based on the Naive Bayes assumption of statistical independence of word occurrence across a document. Our algorithm, takes into account cross-category word occurrence in deciding the class of a random document. Moreover, instead of estimating word co-occurrence in assigning a class, we estimate word contribution for a document to belong in a class. This approach outperforms other statistical classifiers as Naive Bayes Classifier and Language Models, as it was proven in our results.

## Keywords

language models; Naive Bayes classifier; single-label document classification/categorization; statistics

## 1. Introduction

Text classification or text categorization is the task of assigning a random document to a class (single-label classification) or a number of classes (multi-label classification) retrieved from a pre-defined set of possible categories. A special case of single-label classification is binary classification, where the systems choose between two possible classes. Text classification may be applied to numerous areas such as email spam filtering (binary classification) and context identification (multi-label classification) among others. Numerous approaches have been proposed to achieve such a task including, among others, statistics, vector space models, artificial intelligence, decision trees and rule-based methods. One of the simplest approaches is statistical classifiers. The initial assumption of statistical classifiers is the exploitation of observed features that are present in a document, such as word or character occurrence. Statistical classifiers, despite their simplicity or naïve assumptions are proven to achieve great results, outperforming in many cases more complex algorithms on a speed-efficiency trade-off [1]. Another positive characteristic is their speed of execution as opposed to other more complex approaches (SVMs), which allows them to be used for real-time applications [2]. These are the reasons why we chose to research statistical classifiers in general, as part of the work to be presented is based on statistics

---

## Corresponding author:

Athanasios G. Malamos, Technological Educational Institute of Crete, Stavromenos, Heraklion Crete, Greece.  
Email: [amalamos@epp.teicrete.gr](mailto:amalamos@epp.teicrete.gr)

and is part of a larger project for real-time corpus management. Work on statistical methods for text classification such as NBC has been undertaken with significant results regarding its simplicity and efficiency [3, 4], while statistical learning models have often initiated interest either as an autonomous statistical approach [5, 6, 7] alternate to NBC or in conjunction to NBC to achieve better results [8]. Apart from statistical methods another common approach is the definition of vector space models, utilizing algorithms such as k-Nearest Neighbor (kNN) [9]. These algorithms try to identify the similarities of random documents based on a 2D representation of training data either by approximating similarity based on proximity information of a random document through its pre-classified neighbors (kNN), or by visualizing a 2D space split by lines, planes or hyperplanes, denoting the classes a random document may belong to, and try to fit the document into the most appropriate class based on word similarity through vector distance (Rocchio algorithm)[10]. Other approaches include artificial intelligence classifiers (Neural Networks) as in [11]. Latest approaches in the area of statistical document classification introduce the use of keyphrases as in [12]. A keyphrase according to the authors is a set of words commonly found in pre-classified documents, if this keyphrase is apparent in one or few classes. The authors try to identify the existence of keyphrases in random documents, comparing them either on a document level (to identify the similarity between the random document and any pre-classified document), or on a class level (to identify the similarity between the random document and a class).

A common approach shared by all these classification methods includes a training process, where the corresponding systems use a training example set of words or documents already classified into their according classes, and a test dataset, where the efficiency of the systems is estimated, after the training phase. Online corpora to assist in that direction exist, such as the TREC [13] and Reuters [14] corpora for the English language.

In this paper, we deal with single-label, newspaper article classification. We have developed a statistical classifier for Greek language, based on Naive Bayes assumption and cross-category word co-occurrence. In our algorithm, we compute the probability the document belongs to that category given a word belonging to a specific category. Instead of assuming co-existence of words in a document (Naive Bayes assumption) we calculate the Expected Value of a random document to exist in a certain category when a specific word occurs in one category. The Expected Value of the document to belong to a certain category is based on the fact that words may occur in more than one categories, and is calculated through a statistical weight function.

The rest of the paper is organized as follows: in Section 2 we provide insight on statistical classification methods and primarily on Naive Bayes Classifier and Language Models, while in Section 3 we present our approach which significantly differs from both NBC and LM. In Section 4 we provide the algorithms developed, while in Section 5 we present evaluation results from the experiments we run, and comparisons with both Naive Bayes Classifier and Language Models implementations. Finally, in Section 6, we conclude with remarks on our algorithms and future work.

## 2. Background on statistical classifiers

Single-document classifiers have been of utmost importance in the area Machine Learning, mostly due to their ease of use, simplicity and efficiency. Two of the most important statistical algorithms for classification are Naive Bayes Classifier and statistical Language Models. Both of these algorithms try to extract statistically important information from a random document, and through a training process try to fit a random document to one of the accepted categories. These approaches will be used as comparative algorithms for the evaluation of our approach.

### 2.1. Naive Bayes Classifiers

Naive Bayes Classifiers are supervised learning classifiers, based on the Bayes theorem with strong independence assumptions on feature occurrence in a random document. This implies that the occurrence of each feature (a word in our case) in a document contributes independently to the potential class of the document. Consider a set of classes  $C$  and a set of features  $X = (x_1, x_2, \dots, x_n)$ . Classification is based on the maximization of  $P(C|X)$ .

According to Bayes Theorem

$$P(C | X) = \frac{P(X | C) * P(C)}{P(X)} \quad (1)$$

Since  $P(X)$  is the same for any given class in our example set then Formula 1 may be transformed to

$$P(C | X) = P(X | C) * P(C) \quad (2)$$

This is analyzed to

$$P(C|X) = P(C)*P(x_1, x_2, \dots, x_n | C) \quad (3)$$

and since we are referring to independent features  $x_n$  in the feature set  $X$ , then the final formula for calculating the probability a given document with a set of characteristics  $X$  belongs to a category  $C$  becomes

$$P(C|X) = P(C) * \prod_{i=1}^n P(x_i | C) \quad (4)$$

The document, therefore, belongs to the class which maximizes this *a posteriori* probability, often referred to as Maximum a posteriori decision rule (MAP).

Naive Bayes Classifiers have been used extensively in document classification, either as a baseline classifier with which one may compare (almost every paper in the supplied bibliography compares with Naive Bayes Classifier), or through extensions on this initial representation of the algorithm, in order to tackle known problems of the classifier. Extensions on NBC have been proposed by a number of researchers. McCallum and Nigam, for example, in [15,16] and Dai et al. in [17] proposed several extensions on NBC with Expectation–Maximization algorithms, in order to face the costly task of manually labelling an example corpus, by using a small set of labelled corpus and a large set of unlabeled corpus.

**2.1.1 Naive Bayes Classifier efficiency.** The efficiency of the classifier have been outlined in [4] where the author proved through simulation that NBC performs best either on problems with completely independent features, which is expected given the initial hypothesis, or in cases with strongly functionally dependent features. This work also underscores the fact that the algorithms efficiency is lower in cases with weakly dependent features. Moreover, the authors in [3] tried to find inherent problems of Naive Bayes Classifiers and correct them in order to achieve better results. Thus, they found that NBC is bias-prone if the training sets used are uneven.

## 2.2. Language models

Another statistical approach extensively used in text classification is Language Models. Language Models [8, 18] are based on word co-occurrence. They evaluate this co-occurrence by assigning a probability to a sequence of words, by computing its probability distribution. When referring to document classification, a language model is associated with a document in an example set and the random document is evaluated according to the similarity with the language model. Due to the fact that it is not always possible to evaluate the language model in text corpora due to the large number of words that may constitute the language model, an n-gram approach may be followed. In an n-gram language model, the probability of the observation of a sentence  $W = (w_1, w_2, \dots, w_k)$  can be calculated as

$$P(W) = P(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i | w_1, w_2, \dots, w_{i-1}) \approx \prod_{i=1}^k P(w_i | w_{i-(n-1)}, \dots, w_{i-1}) \quad (5)$$

Since, it is considered that the probability of the occurrence of word  $i$  of the sentence in the context history of the preceding words can be approximated by the probability of observing it in the previous  $n-1$  words. Language models have been used as an alternate approach to NBC in an attempt to evaluate the statistical dependence of words that may be apparent in a sentence. An estimation of the maximum likelihood estimate of the n-gram probabilities may be given by the observed frequency

$$P(w_i | w_{i-(n-1)}, \dots, w_{i-1}) = \frac{(w_{i-(n-1)} \dots w_i)}{(w_{i-(n-1)} \dots w_{i-1})} \quad (6)$$

The simplest form of Language model is the unigram language model, where all conditional information is disregarded and each term is considered independently. A unigram model according to (5) is:

$$P_{n=1}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i) \quad (7)$$



Other commonly used language models engage bigrams (8) and trigrams (9), in an attempt to not only evaluate the existence of words in a document inferring dependency between their appearance, but also, to evaluate the order of their appearance in the document as critical to identifying the context of a random document. Thus, San Francisco as a bigram that may be commonly found in random documents will be evaluated according to the dependent probability of San preceding Francisco.

$$P_{n=2}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i | w_{i-1}) \quad (8)$$

$$P_{n=3}(w_1, w_2, \dots, w_k) = \prod_{i=1}^k P(w_i | w_{i-2}, w_{i-1}) \quad (9)$$

One of the first research works in Language Models for IR was undertaken by Ponte and Croft presented in [7], where they proposed a language modelling technique for classification tasks, and carried out experiments that proved that Language Models produced better results than standard tf.idf weighting techniques. Examples of work in language models have been undertaken by Peng et al. [8] where the authors tried to enhance Naive Bayes Classifier with Language Model characteristics, in order to overcome the statistical independence of NBC. A direct link between NBC and Language Models has been observed by the authors vividly stating that unigram Language Model classifier with Laplace smoothing actually corresponds to the traditional NBC. In addition to that, the authors experimentally prove that bigram classification performs better than NBC. Language Models infer ordered sequence of words as they appear in a sentence, in order to estimate the statistical dependence of the word sequence occurrence. Language models have been used as a means to estimate unordered word occurrence by Srikanth and Srihari in [5], where they proposed three approaches on estimating unordered word occurrence (referred to as biterm as it consists of two words) on random documents: through the average of the components of a bigram language model, the term frequencies of both words to the occurrence of the first observed word, and the term frequencies of both words to the minimum of the term frequencies. Their experiments showed that Language Models may fail to be as effective as unordered word n-grams observation.

### 3. Our approach

The Naive Bayes assumption of statistical independence states that words in a document are statistically independent with regards to their appearance, and has been proven to work surprisingly well, considering the initial false assumption. However, NBC classifier suffers from bias over unequal in length classes (as proven by Rish in [4]). In our approach, we have developed a single-label, newspaper article supervised trainable classifier for the Greek language that uses a normalized approach in assigning a random document to a class. The system takes into account term frequency, along with the size of each class. Therefore, each class component has a normalized weight coefficient, participating in the final classification step. The main difference with common approaches, such as the NBC, is that we did not utilize a product methodology in computing the importance of words in a document to extract the category a document belongs to, but rather the sum of each of the weight coefficient of each word for a category. Our approach considers that each word is statistically independent in occurring in a random document, and instead of trying to identify word co-occurrence as in NBC and LMs (computing the product of each probability of word occurrence, therefore searching for word co-occurrence), we estimate the contribution of each independent word to the class of the document. Given a word in a document, we consider the probability of the document to belong to a specific category featuring that word, and we further estimate the expected value (mean probability) of the contribution of the entire word set of the document to each one of the available categories. A direct outcome from using the sum of probabilities rather than the product is that, while we were initially targeting to classify a document in exactly one of the available categories (single-label classification), the system was able to identify more than one potential category (multi-label classification). Another outcome is that error is not propagated in sum as quickly as in the product, thus making the algorithm less susceptible to noise.

Another assumption we made in developing our system dealt with what words should be considered as important in our system. We consider only nouns to be important in identifying the context of a sentence or document rather than other parts of speech [18]. This has also been verified in work by Bouras and Tsogkas in [19], where the authors implied that nouns extraction greatly assists in Classification and Summarization tasks. The reason for that is that we consider the nouns to hold the essence of a sentence, while verbs and other parts of speech operate either complementary to the meaning or show action between nouns. This is achieved through a removal of words as articles, pronouns, propositions and adverbs, while also the text is processed during the training phase to ignore numeric and alphanumeric words. Verbs and participles are ignored according to their unique word endings, where applicable. Thus, a sentence for our system is

denoted by the stems of the nouns the beginning of a paragraph and a dot/question mark/exclamation mark or two dot/question marks/exclamation marks.

Our initial thoughts formulated the following problem: Let  $i$  be a random noun, and  $D$  a random document featuring word  $i$ , and  $j$  a category the document may belong to. We are trying to compute what is the possibility for document  $D$  to belong to category  $j$  given that word  $i$  appears in  $D$ .

In order to compute this we assign as  $w_{i,j}$  as the weight of word  $i$  in category  $j$  computed as

$$w_{i,j} = \frac{|tf(i,j)|}{\sum_{i=0}^n |tf(i,j)|} \quad (10)$$

where  $tf(i,j)$  the term frequency of word  $i$  in category  $j$ , and  $n$  the total number of unique words comprising category  $j$ .  $w_{i,j}$  in this context denotes the importance or contribution of noun  $i$  in category  $j$ . By dividing with the total number of the noun term frequencies of category  $j$ , we form a normalized weight factor for nouns in that category, in order to overcome NBC bias. This formula is influenced by the Term Frequency (TF) used in Information Retrieval algorithms (the first parameter of tf.idf algorithm). This approach also enables us to properly estimate the similarity of a random document to a category, since a great number of significant words of a category in a random document (high-weighted word observation), denotes greater similarity between the document and the category. The similarity factor of our approach is based on the probability of a random document  $D$  belonging in category  $j$ , if word  $i$  is present in document  $D$ . Given that word  $i$  is assigned a weight per class  $j$ , then this probability is calculated by

$$p_{i,j} = P(D \in j | i \in D) = \frac{w_{i,j}}{\sum_{j=0}^n w_{i,j}} \quad (11)$$

where  $n$  is the total number of categories the system can identify. This metric takes into account the cross-class importance of the words. This algorithm strongly resembles the Inverse Document Frequency metric used in Information Retrieval (the second parameter of the tf.idf algorithm).

The evaluation criterion that denotes a document belonging to a category is called similarity factor ( $sf$ ) and is calculated by

$$sf(D, j) = \frac{\sum_{i=0}^{Dwords} \frac{w_{i,j}}{\sum_{j=0}^m w_{i,j}}}{Dwords} \quad (12)$$

where  $m$  is the total number of categories available and  $Dwords$  the word count of document  $D$ . Since  $sf$  is computed on the observed set of nouns extracted by the document, and not all words appear in category  $j$ , the contribution for each word present in the document is either 0 if the word is not present in category  $j$  or calculated according to (11). Dividing by  $Dwords$  gives us the Expected Value for each category  $j$ . The system decides that the document belongs to the category which maximizes the similarity factor  $sf$

$$D \in j \Leftarrow \operatorname{argmax}(sf(D, j)) \quad (13)$$

where  $\operatorname{argmax}(sf(D, j))$  is the maximum similarity factor of Document  $D$  in categories  $j$ .

For example, our system can identify six potential categories of articles namely Business and Finance, Politics, Culture, Sports, Technology and Health. A word that is part of all six datasets is the word 'πολιτικ' – the stem of word politics or politician. Applying formula (10) on word stem 'πολιτικ' based on information gathered from the training phase, we calculate the weight per category to be as shown in Table 1.

These results indicate that given a document containing only the word 'πολιτικ', then it would be classified in category Politics since it holds the greatest similarity factor among every category – the weight of the word being the similarity factor of the document to each category in this case.

In the next section we provide our methodology in pseudo-code and analyze each step of the classification process.



**Table 1.** Weight of word “πολιτικ” in every category

Category	Occurrence	Total Category Words	Weight
Business and Finance	196	59777	0.149275137
Culture	179	26932	0.302586779
Health	1	3073	0.014815045
Politics	664	63218	0.478181589
Sports	11	25429	0.019693773
Technology	12	15412	0.035447676

## 4. Methodology

We have developed a system based on the aforementioned approach. The system is based on a learning step (made up by a stemming step and a training step) and a test step, while information undergoes a preparatory suffix stripping phase. Initially every document undergoes a stemming procedure. This produces sets of stems of words that are provided as input to the system. The stemming procedure is also responsible for isolating nouns from other parts of speech. The second step of the system is the training phase where the system is provided with the example set that is going to be used to evaluate word significance per available class. The third step is the classification phase, where the system takes as input a random document and classifies it into one or more of the available categories. The system is intended to be used as a newspaper article classifier for Greek language.

### 4.1. Stemming step

Stemming is the process of identifying the stem of a word (the part that does not alter in the different forms that a word may be found in a random document) from its suffix (ending). Initial work in the area was undertaken by Porter in [20] where he proposed a system for suffix stripping for English language based on grammatical features. This work was very important as it has been used in a number of Machine Learning applications as document classification – where Scott and Matwin in [21] vividly state that it is almost always used- and summarization.

The initial work for stemming in Greek language was undertaken by Kalamboukis [22], who adapted Porter’s work for Greek language, based on greek grammatical features. Kalamboukis work included the gathering of all potential endings of suffixes of words and the extraction of the stem of the word. His work was used as a basis for our research on the area of stemming. We decided to develop a stemmer [23] that would extend Kalamboukis approach by performing minimal part of speech tagging work. There are two main reasons for that, the first one being the fact that we only wanted noun identification and the second that no efficient Greek POS tagger exists. The second point is very crucial to our approach, since while our stemmer may include non-noun information in the data sets it never fails to correctly identify a Greek noun. All other surplus data is either ignored in the classification stage, or is already existent in the training phase and therefore has already affected the importance factor of each word. On the other hand, failing to correctly identify a noun (as in [24], although the system is trainable and therefore may be suitable for the task in a costly manner) breaks the second assumption we made in the previous section. The stemmer includes a stopwords removal phase (using a set of stop words for articles propositions, pronouns and adverbs) as well as verb and noun endings sets. Our stemmer may include insignificant data in the final system, mostly due to existence of Greek language particularities – common suffixes between nouns and adjectives (e.g. ναυτικός – sailor and τακτικός – tactical, the first being a noun while the latter an adjective) or nouns often used as adverbs (e.g. αλήθεια – truth as a noun and really as an adverb). Still, these particularities on a more generalized aspect do not constitute a major problem on the overall efficiency of the system. The stemming step algorithm operates as in Figure 1.

The final table  $w_p$  stands for the document to be included in training or classification.

### 4.2. Training step

The training step is responsible for building up the dictionaries of words used in classification. Each newspaper article in each category is stemmed and the according article noun-table is used to create the weighted category dictionary. Each word is assigned a normalized weight according to the number of nouns present in each category. The training module algorithm is depicted in the Figure 2.

```

1. Let random document D
2. Split D in words, forming wD
3. For each l in wD
4. If l is an Article Then remove l End If
5. If l is a Preposition Then remove l End If
6. If l is a Pronoun Then remove l End If
7. If l is an Adverb Then remove l End If
8. If l is a Verb Then remove l End If
9. If l is a Noun Then
10. If l is a Participle Then remove l Else keep l End If
11. End If
12. End For
13. Gather updated table of words wD

```

**Figure 1.** Stemming algorithm

```

1. For each category j formed by documents Dj
2.   Create an empty dictionary for category j (dictj), where each dictionary entry is a quadruple (word, cat_freq, weight, probability) *
3.   For each document D of Dj
4.     Stem D and acquire word table wD
5.     Create a vector vD of couples (word, doc_freq) **
6.     For each couple c of vD
7.       If dictj (the dictionary of j category) contains c.word Then
8.         Update occurrences (.cat_freq) of the corresponding couple of dictj
9.       Else
10.        Append a new quadruple to dictj initialized as (c.word, c.doc_freq, 0, 0)
11.      End If
12.   End For
13. End For
14. Compute Sj as the sum of all.cat_freq (frequencies) in dictj
15. For each entry e of dictj
16.   Compute e.weight as e.cat_freq divided by Sj
17. End For
18. End For
19. For each category j formed by documents Dj
20. For each entry e of dictj
21.   Compute the sum of weight items of every dictionary entry of every category that has.word item equal to the present word entry (e.word) ***
22. Compute e.probability by dividing e.weight with the sum
23. End For
24. End For

```

**Figure 2.** Training Module Algorithm

\* cat\_freq is the total number (sum) of occurrences of word inside all documents of category j, weight is computed according to formula (10), at step 16, probability is computed according to formula (11), at step 22.

\*\* doc\_freq is the frequency of word inside D

\*\*\* step 21 computes the denominator of formula (11)

### 4.3. Classification step

The classification step is responsible for assigning a random document in one category. Each random document is stemmed and the resulting noun table is checked in each category to form the similarity factor. The category a document belongs to is defined by the maximum similarity factor. The classification step algorithm operates as in Figure 3.

## 5. Evaluation results

We have tested our system against two statistical algorithms, NBC and statistical Language Models, as provided by Mallet [25] and Lingpipe [26] natural language processing toolkits respectively. The statistical Language Models utilized included a 6-gram Language Model and a trigram Language Model. Both Mallet and Lingpipe provide a Java development API and were included in a test-bed evaluation application, along with our algorithm.

1. Let random document D
2. Stem D and acquire word table wD
3. For each category j
4. Initialize the similarity measure of document-category sfDj ( $\text{sfDj} \leftarrow 0$ )
5. For each word i of wD
6. Locate the entry e of dictj that corresponds to word i
7. If the location is successful (dictj has an entry for the given word i) Then
8. Add to similarity measure of document-category sfDj the word probability for this category  $\text{pi.j}$  ( $\text{sfDj} \leftarrow \text{sfDj} + \text{e.probability}$ )
9. End If
10. End For
11.  $\text{sfDj} \leftarrow \text{sfDj} \div \text{sizeof}(wD)$
12. End For
13. Return j that has the maximum sfDj

**Figure 3.** Classification Step

### 5.1. Corpus profiles

The training and test dataset were randomly gathered from online Greek newspapers and was initially classified according to a unique classification scheme: Business and Finance, Culture, Health, Politics, Science and Technology, and Sports. The training corpus was comprised of 1015 articles and the test corpus of 353 articles. The corpora were randomly gathered by a number of newspapers in order to include a great number of authoring styles and vocabulary. The training corpus and test corpus were gathered over a period of a semester. They are both available at: [http://www.medialab.teicrete.gr/classification\\_corpus.rar](http://www.medialab.teicrete.gr/classification_corpus.rar). Each article in the training corpus underwent the stemming procedure and the resulting categories had the characteristics as depicted in Table 2.

As it can be observed, each category is formed from around 3500 to 7100 unique words, except for Health class which is comprised of only 1181 unique nouns. The total number of words shows the total number of nouns included in each category (unique words times word frequency). Each category depicts its own average word occurrence and according word weight. The average word weight ( $w_{i,j}$ ), excluding Health class, is around 0.000142 and 0.000278 depending on number of total words, while in Health class the average weight is 0.000847, denoting that a word present in Health class is 3 to 5 times more important than this word in any other domain.

### 5.2. Experiments

All classifiers were provided as input stems of words in the training step and stems of nouns in the sentences, in exactly the same manner. Thus, a number of interesting results were acquired. The complete results (Table 3) showed that our algorithm outperformed both Naive Bayes Classifier, 6-gram and trigram Language Model.

In Table 3, positive results are considered to be the ones where the algorithms managed to correctly match the human assigned class, whereas negative results are considered the ones that the algorithms failed to correctly identify. Thus, as it may be seen our algorithm outperformed both Naive Bayes Classifier and Language Models. More specifically, our classifier achieved a percentage of 92.35% correctly identified articles, while all other algorithms achieved well below 90%.

**Table 2.** Nouns per category

Category	# of unique words	# of total words	Average word occurrence	Average weight
Business & Finance	5686	59777	10.51	0.000176
Culture	5750	26932	4.68	0.000174
Health	1181	3073	2.60	0.000847
Politics	7059	63218	8.96	0.000142
Science & Technology	3593	24429	6.80	0.000278
Sports	4696	15412	3.28	0.000213
Totals	27965	192841	6.139254653	0.000304909

**Table 3.** Overall Classification Results

	Our Classifier	NBC	LM-6	LM-3
Positive				
#	326	302	284	294
%	92,35	85,55	80,45	83,29
Negative				
#	27	51	69	59
%	7,65	14,45	19,55	16,71
Totals				
#	353	353	353	353
%	100	100	100	100

### 5.3. Results per category

In the following table (Table 4), we will try to project per category efficiency of our algorithm when compared to NBC and LMs, for single-labeled documents, as some interesting results may be extracted.

First of all, our algorithm produced for the given test corpus better (or as good results in some cases) as NBC and LM. The results between the algorithms are comparable among all categories, except for Health class. Health class is made up of the smallest dictionary of all six categories, while it contains a number of words similar to Science & Technology class. Our algorithm correctly classifies 27 out of 43 Health articles in the test corpus (almost 63%), as opposed to NBC and LM which face serious problems (23%, 12% and 30% of the total Health articles correctly classified). A reason for that is that while NBC is indifferent to word co-occurrence it also treats every class as independent to one another trying to maximize the highest scoring category. This is the characteristic that tends to create bias towards a class with larger datasets as observed in [4]. Health class in our test was comprised of the smallest dataset, sharing common words with Science and Technology, the latter being more than twice as big as Health corpus. Therefore, NBC tended to classify these documents incorrectly. Contrary to that, our initial target in the system was to treat each domain as equally probable, eliminating any bias. This was achieved through the weight calculation formula used to estimate word contribution to a category, through

**Table 4.** Classification Results per Category

Categories	Our Classifier		NBC		LM-6		LM-3	
	#	%	#	%	#	%	#	%
Business & Finance								
Positive	66	95,65	63	91,30	59	85,51	60	86,96
Negative	3	4,35	6	8,70	10	14,49	9	13,04
Totals	69	100,00	69	100,00	69	100,00	69	100,00
Culture								
Positive	54	96,43	54	96,43	53	94,64	52	92,86
Negative	2	3,57	2	3,57	3	5,36	4	7,14
Totals	56	100,00	56	100,00	56	100,00	56	100,00
Health								
Positive	27	62,79	10	23,26	5	11,63	13	30,23
Negative	16	37,21	33	76,74	38	88,37	30	69,77
Totals	43	100,00	43	100,00	43	100,00	43	100,00
Politics								
Positive	47	97,92	47	97,92	46	95,83	42	87,50
Negative	1	2,08	1	2,08	2	4,17	6	12,50
Totals	48	100,00	48	100,00	48	100,00	48	100,00
Science & Technology								
Positive	73	93,59	70	89,74	62	79,49	69	88,46
Negative	5	6,41	8	10,26	16	20,51	9	11,54
Totals	78	100,00	78	100,00	78	100,00	78	100,00
Sports								
Positive	59	100,00	58	98,31	59	100,00	58	9,31
Negative	0	0,00	1	1,69	0	0,00	1	1,69
Totals	59	100,00	59	100,00	59	100,00	59	100,00

the observed frequency. Moreover, we consider that a word may exist in more than one category, with different weights per category. Therefore, it is important to estimate the overall importance of this word not only on one category but also on a cross-category level. These two estimates tend to produce less biased results on small datasets that share common words with larger datasets, as Health class.

#### 5.4. Tests with ambiguous data

During our tests, we also observed that our algorithm produced category results that in some cases were ambiguous, especially for articles that their manual classification by the newspapers was ambiguous, since their content was semantically shared between 2 or 3 thematic areas. For example, in our test article a83 originally classified in Politics by the newspaper, dealt with the cultural effects of the elections on a country through history. This implied that while Culture is the primary class for that article, Politics could also be a potential category. In fact our classifier, identified both categories with Culture having an  $sf$  value of 0.2263 and Business and Finance having an  $sf = 0.2214$  while other classes'  $sf$  were in the region of 0.0067 to 0.1469. Motivated by that, we tried to verify how the system would operate with ambiguous input data, in order to check its robustness. We gathered 23 newspaper articles that could not be classified into one category. We assigned two categories per article in order of similarity (e.g. Business and Finance/Politics denoting that this article fits into Business and Finance primarily and Politics secondarily) and run the experiment for these articles.

The results acquired are shown in Table 5.

In this case, an extra selection from positive and negative is used, namely ambiguous. In this case Positive stands for correct classification of the document in all classes in order of classification similarity, negative failure to categorize the document in any of the classes it belongs to and ambiguous either successfully categorizing into one of the categories of the document, but not all of them, or successfully categorizing into all categories but with different order from the one supplied by the human classifier. As we observe, the efficiency of the classifier correctly identifying the classes of a document is similar to the one of single-label classification. Therefore, this hints that the system could be used for multi-label classification as well, since the results are very promising. However, this is beyond the scope of current work, as the intention of this paper was to exclusively deal with single-label classification.

## 6. Conclusion

In this paper, we presented a statistical approach for newspaper article classification that significantly outperforms both NBC and LMs. This approach is based on the same fundamentals as NBC, yet instead of utilizing the product of each NBC feature, our approach uses the sum of each feature probability. This reduces error propagation and bias towards specific categories, constituted by large datasets. This approach, also, enables both single-label and multi-label classification, as the algorithm engages a normalized similarity factor, that empirically was found to approximate effectively cross-class classification. Another important remark from our experiments regarding classification was that Language Models efficiency increased when setting the n-gram size to three from six. Potentially their performance will increase if we use a bigram, but still it is not expected to outperform our approach. Future work includes the evaluation of the system on multi-label classification tasks, driven by the fact that example results on a very small corpus supplied promising hints on the overall performance of the classifier. Yet, this is beyond the scope of this paper, and the algorithm has to be evaluated with statistical multi-label algorithms.

**Table 5.** Classification with ambiguous input data

	Our Classifier
Positive	
#	21
%	91,30
Negative	
#	0
%	0,00
Ambiguous	
#	2
%	8,70
Totals	
#	23
%	100



## References

- [1] S.B. Kotsiantis and P.E. Pintelas, Increasing the classification accuracy of simple bayesian classifier, *Lecture Notes in Artificial Intelligence, AIMS 2004* 3192 (2004) 198–207.
- [2] G. Tsoumakas, I. Katakis and I. Vlahavas, A review of multilabel classification methods, *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery* (2006) 99–109.
- [3] J.D.M. Rennie, L. Shih, J. Teevan and D. R. Karger, Tackling the poor assumptions of Naive Bayes Text Classifiers, *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)* (Washington, DC, USA, 2003).
- [4] I. Rish, An empirical study of the Naive Bayes Classifier, *Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, 41–6 (Seattle, USA, 2001).
- [5] M. Srikanth and R. Srihari, Bitern language models for document retrieval, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 425–6 (Tampere, Finland, 2002).
- [6] W.B. Croft, Language models for information retrieval, *Proceedings of the 19th International Conference on Data Engineering* (Bangalore, India, 2003).
- [7] J.M. Ponte and W.B. Croft, A language modelling approach to information retrieval, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 275–81 (Melbourne, Australia, 1998).
- [8] F. Peng, D. Schuurmans and S. Wang, Augmenting Naive Bayes Classifiers with Statistical Language Models, *Information Retrieval* 7 (2004) 317–45.
- [9] C.D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge: Cambridge University Press (2008).
- [10] J. Rocchio, Relevance Feedback in Information Retrieval, *The SMART Retrieval System: Experiments in Automatic Document Processing* (1971) 313–23.
- [11] B.D. Ripley, Neural networks and related methods for classification, *Journal of the Royal Statistical Society. Series B (Methodological)* 56 (1994) 409–56.
- [12] N. Karanikolas and C. Skourlas, A parametric methodology for text classification, *Journal of Information Science* 36(4) (2010) 421–42.
- [13] TREC Corpus, <http://trec.nist.gov> (last accessed 30/7/2010).
- [14] Reuters Corpus, <http://trec.nist.gov/data/reuters/reuters.html> (last accessed 30/7/2010).
- [15] A.K. McCallum and K. Nigam, A comparison of event models for Naive Bayes Text Classification, *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 41–8, ( Winsconsin, USA, 1998).
- [16] K. Nigam, A.K. McCallum, S. Thrun and T. Mitchell, Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning* 23 (2000) 103–34.
- [17] W. Dai, G.R. Xue, Q. Yang and Y. Yu, Transferring Naive Bayes Classifiers for Text Classification, *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 540–5 (Vancouver, Canada, 2007).
- [18] M. Galley and K. McKeown, Improving word sense: Disambiguation in lexical chaining, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 1486–8 (Acapulco, Mexico, 2003).
- [19] C. Bouras and V. Tsogkas, Improving text summarization using noun retrieval techniques, *Lecture Notes in Computer Science* 5178 (2010) 593–600.
- [20] M.F. Porter, An algorithm for suffix stripping, *Program* 14(3) (1980) 130–7.
- [21] S. Scott and S. Matwin, Feature Engineering for Text Classification, *Proceedings of the 16th International Conference on Machine Learning* (1999) 379–88.
- [22] T.Z. Kalamboukis, Suffix stripping with modern Greek, *Program: Electronic Library and Information Systems* 29(3) (1995) 313–21.
- [23] G. Mamakis, A.G. Malamos, Y. Kaliaktos, A. Axaridou and J.A. Ware, An algorithm for automatic content summarization in modern greek language, *Proceedings of ICICT '05* (Cairo, Egypt, 2005).
- [24] AUEB Greek POS Tagger, (2010), [http://nlp.cs.aueb.gr/software\\_and\\_datasets/AUEB\\_Greek\\_POS\\_tagger.tar.gz](http://nlp.cs.aueb.gr/software_and_datasets/AUEB_Greek_POS_tagger.tar.gz) (last accessed 20/7/2010).
- [25] A.K. McCallum, *MALLET: A Machine Learning for Language Toolkit*, <http://mallet.cs.umass.edu> (last accessed 20/7/2010).
- [26] Alias-I, *LingPipe 4.0.0*, <http://alias-i.com/lingpipe> (last accessed 20/7/2010).

# AN ALGORITHM FOR AUTOMATIC CONTENT SUMMARIZATION IN MODERN GREEK LANGUAGE

*Georgios Mamakis*

*Department of Applied Informatics and Multimedia  
School of Applied Technology  
Technological Educational Institute of Crete  
gmamakis@epp.teicrete.gr*

*Athanasios G. Malamos*

*Department of Applied Informatics and Multimedia  
School of Applied Technology  
Technological Educational Institute of Crete  
amalamos@epp.teicrete.gr*

*Yannis Kaliakatsos*

*Department of Electronics  
School of Applied Technology  
Technological Educational Institute of Crete  
giankal@chania.teicrete.gr*

*Anastasia Axaridou*

*Department of Applied Informatics and Multimedia  
School of Applied Technology  
Technological Educational Institute of Crete  
a\_axaridou@yahoo.gr*

*Andrew Ware*

*School of Computing  
University of Glamorgan  
jaware@glam.ac.uk*

## 1. INTRODUCTION

From early on in the history of Information Technology, one of the topics that attracted interest from researchers around the world is developing an automatic procedure for extracting or generating content summarization of any given document. By automatic content summarization, we refer to producing a summary based either on sentence extraction or sentence generation, using the topic discussed in a document as the guide in developing such a methodology. In this paper, we propose an algorithm for extracting a summary in documents written in Greek language. We have defined a 4-step procedure capable of extracting summarization based on the topic of a document, by utilizing stemming with grammatical rules, semantics and a

proposed methodology for the document summarization extraction itself. The rest of the paper is organized as follows: In chapter 2, we provide background information on the research undertaken that is applicable on the area of document extraction and more specifically on the steps of the proposed algorithm, in chapters 3-6 we present the methodology we utilized to generate this algorithm as well as an example of the algorithm.

## **2. BACKGROUND**

A lot of research has been undertaken in the area of text summarization. The initiation of the interest in this area, was the assumption stated in (Luhn, 1958) that the repetition of a word in a scientific document, is distinctive of the subject of the document and that therefore a statistical analysis of each word in a document may be enough not only to verify the subject of the document, but extract those sentences containing words who possessed a greater occurrence ratio in the document. This assumption was refined in (Baxendale, 1958) and (Edmundson, 1969), stating that a sentence's relative position and a structural analysis respectively, would offer better results. Since then, a number of approaches have been researched both for summary generation and for summary extraction on single- or multi-document systems. In (Teufel and Moens, 1997), for example, the authors proposed an algorithm for document extraction based on rhetorical features present in scientific documents. Such rhetorical features are word sequences that represent a specific meaning to be presented. "Therefore...", for example, is most probably used to point out that the solution to a given problem is to be discussed. (Barzilay and Elhadad, 1997) proposed a system making use of WordNet semantic lexicon and the idea of lexical chains (sequences of word occurrences or synonyms in a document and their sentence distance), to provide a system for automatic text summarization, using a statistical analysis. A similar idea was introduced by (Benbrahim and Ahdad, 1995), where they used repetition schemes between words and their derivatives in a document to statistically extract a summarization. (Leskovec et al., 2004) proposed the use of semantics as a tool for extracting document summaries. In this paper, the use of semantic domains was instantiated as a primary key to solving the heterogeneity in the vocabulary of the document, thus assisting the automatic extraction of summaries.

## **3. METHODOLOGY**

All languages share some characteristics. First of all, in order to define a language one needs to define an alphabet. According to (Charras and Lecroq, 2004), alphabet is a finite set of characters used to construct words, which are the basis of a language. The second characteristic identified as vital for the existence of a language is a word. A word is a meaningful sequence of characters belonging to the alphabet of a language. The set of these words



recognized by a language is called vocabulary. The rule enabling the creation of such meaningful words and further utilization of them in a text is called a grammatical rule. Apart from the aforementioned, another characteristic of languages is the ability to form sentences by putting words belonging to their vocabulary, one after the other and produce valid meanings. A process which defines the structure of these sentences is called a syntactic rule. These rules described are not just characteristics used in language construction, but must also be considered when producing an algorithm capable of extracting content summaries. Content relations between word meanings, may also interfere with the algorithm, especially in multi-modal languages as Greek language. Greek grammar is based on a great variety of rules that apply specifically for each grammatical element. By grammatical element, we refer to nouns, adjectives, verbs, articles. For the rest of the paper, we refer to grammatical elements, also, as word type. These grammatical elements, as opposed to the English language may appear in more than one form in a text. The suitable form for each word is selected according to the meaning one may wish to express. At this point it is necessary to know the grammatical rules that specify how a word would be transformed in order to be used in a sentence. The possible transformations of a word depend on the word type it belongs. In Greek language, the possible word forms of a word are affected by a number of factors as the word type, its syntactical use in a sentence, and the grammatical rules it corresponds to not only for that specific word type, but for the subcategory of the word type it belongs to. In English language the factors that affect possible word forms, for example, are only the word type it belongs to and the corresponding rules for that word type, since syntactical position and subcategories are irrelevant or non-existent for the English language structure scheme. Therefore, the first preparatory part in an attempt to create a summarization algorithm is stemming for Greek words using grammatical features. The second preparatory part for the operation of the algorithm is the creation of semantic domains, crucial for the document's topic validation, as well as the utilization of a semantic lexicon that would offer relationships between words other than their presence in a domain. The third and fourth part of our methodology is the two-stepped algorithm that initially verifies the topic of the document and subsequently extracts the document summarization, based on Greek nouns.

### **3.1 Stemming for Greek language**

In order to produce a stemming algorithm for Greek language, it is to verify the structure of Greek words and the respective word ending defined for each word form. We will present two cases for Greek nouns and Greek verbs and evaluate the encapsulation of such word types in producing a document extraction algorithm.

#### **3.1.1 Grammatical and syntactical rules for Greek nouns**

Nouns are divided into 3 categories expressing genders: masculine, feminine, neutral. Each noun may have a form in either all 3 genders, in 2 or in one of them. The translation of the word “king”, for example, in Greek is «βασιλιάς», and it is of masculine gender. It has a feminine gender, meaning “queen”, which stands for «βασίλισσα». It also has a neutral gender, which can not be translated in English in one word meaning the “young king” and is represented as «βασιλόπουλο». Another grammatical rule that interferes with noun construction is its use in a sentence. (A noun as a subject is always in nominative case, while as an object, it is always in genitive or accusative case). While in English each word has a specific type that is maintained unchanging regardless if the noun is a subject or an object, in Greek language this does not apply. A number of word forms are defined according its relative syntactic position in a sentence. These forms are created regarding four cases.

- a) When the noun is to be used as a subject to a sentence we use a form called «Ονομαστική» (nominative). In nominative case the noun is formed according to its primary form. Word «οδηγός» (driver) is in nominative case.
- b) When the noun is used to show that a subject belongs to it, the form used is called «Γενική» (genitive). The genitive case of word «οδηγός» is «οδηγού».
- c) When the noun is an object, then we use «Αιτιατική» (accusative) case. The accusative case of noun «οδηγός» is «οδηγό».
- d) When we want to refer to a noun we use «Κλητική» (vocative) case. The vocative case of word «οδηγός» is «οδηγέ».

In some cases the nouns are maintained unchanged. For example, the word forms of word «βασιλιάς» are in turn:

Nominative: βασιλιάς  
Genitive: βασιλιά  
Accusative: βασιλιά  
Vocative: βασιλιά

These word forms apply for feminine and neutral categories, as well.

As one may have noticed from the previous examples, using a noun in a sentence does not differentiate the primary word root structure a lot. A main part of the word remains the same. For this reason, we define as word ending the part of the word those changes when a word is structured according to the aforementioned rules. This part of the word is always the last few letters of it (2-3 letters). This assumption applies to all the nouns used in Greek language. The available word endings vary for each category described.

For masculine nouns the typical word endings in nominative case regardless of punctuation marks are:

-ος, -ης, -ας, -ες, -ους

For feminine nouns the typical endings are:

-α, -η, -ω, -ος, -ου

For neutral nouns the typical endings are:

-ο, -ι, -ος, -ως, -ας, -μα

Another feature that changes word forms are singular or plural numbers. Again the main form of a noun remains the same, while the ending syllable or syllables change. If punctuation marks are taken into consideration the resulting word endings become difficult to handle. Still we can be definite that the main form of the noun remains unchanged. Therefore, nouns may be included in an algorithm for content summarization.

### 3.2 Stemming methodology

In our preparation, we defined an algorithm based on Porter stemming algorithm (Porter, 1980) appropriately adapted for the Greek nouns. The algorithm uses a stop-list of possible word-endings and compares them with words found in a document. If a word does not have one of the possible endings it is not taken under consideration. However, even if a word has one of the accepted endings, then it may be a noun, an adjective or a passive participle. Passive participles have similar endings to nouns in the Greek language but passive participles always end in “μένος”, “μένη” or “μένο” and their respective conjugation. On the other hand adjectives are dealt by the text classification and summarization algorithm (section 5) and the use of domain specific lexicons, which are explicitly consisted of nouns. So, during the nouns comparison and selection procedure, adjectives are disregarded.

The algorithm is as follows:

*Pre-processing stage*

*Create stop-list Sj of noun endings for masculine, feminine and neutral nouns*

*Stage 1 Create the dictionary*

*Let D a document, w the words of a document, e{} an array containing the last letters of the word w and i the number of letters taken from a word, dict the dictionary of nouns dictionary.*

*Begin*

```

For each w in D
For i=1 to 4
If  $S_j = e\{1 \dots i\}$  then           "assume the last i letters of the word"
Add  $w-e\{1 \dots i\}$  to dict         "add the stem of the word in the dictionary"
Else
Disregard the word
Endif
End for
Endfor
End

Stage 2 Calibrate the dictionary to avoid participles
Begin
For each w in dict
If  $e\{3\} = \langle \mu\acute{\epsilon}\nu \rangle$  then      "take the last 3 letters of the stemmed words"
Remove w from dict
Endif
Endfor
End

```

#### 4. UTILIZING SEMANTICS TO EXTRACT DOCUMENT SUMMARY

The second preparatory step taken is the definition and creation of semantic domains that would verify the topic of any given document. Before describing the proposed methodology, let us describe the human process of extracting summaries from a document. First of all, a human must read the whole document. Then, according to his previous experience, he decides which of the available sentences are suitable to be included in the summary. The most interesting feature in the depicted process is the part referring to previous experience. Previous experience typically includes knowledge on the thematic subject of the document. If the human does not know anything about the thematic subject then the process of extracting a summary is far more difficult. Knowledge on the subject implies that the user knows keywords and the value of each keyword in a sentence (how important it is) related to the main subject of the document he is reading. A domain is made up of a word representing a central meaning (e.g. "Politics") and a set of words, and in our case nouns, either directly related to each other or most likely to be found together on a text referring to that meaning (e.g. "politician", "citizen"). The definition of domains of words implies the utilization of semantics. Semantics are word associations based on meaning relations of these words. In order to extract semantic domains, we have used a statistical analysis based on the thematic domain creation methodology proposed by (Baker and McCallum, 1998).

## 5. ALGORITHM FOR DOCUMENT SUMMARY EXTRACTION

The algorithm developed for document extraction consists of two parts: text classification and content summarization. Text classification stage is responsible for classifying the document on a specific domain, while content summarization step is responsible for extracting a summary of a specified length of words, in approximation. The methodology used for text classification is an adaptation for Greek language of portions of the algorithm proposed by (Edmundson, 1969), with semantic enhancements satisfying the limitations imposed by the language itself. The text summarization algorithm follows the statistical procedure initially described by (Luhn, 1958), using keyword based search and statistical analysis with semantics enhancement to create an automatic summarization system.

### 5.1 Text Classification

On the first step, the algorithm scans through the whole document to verify the domain it belongs. This is done with the use of the semantic graphs of each domain. The domain a document belongs to is computed using formula:

$$d = (V \cap D) / (V).$$

where  $d$  is the domain the document belongs to,  $D$  the domain and  $V$  the nouns of the given document. The nouns of each document are estimated with the use of the proposed Greek stemming algorithm. If  $d > 30\%$  then the document belongs to that domain. The limit is arbitrary. However, experience has shown to us that setting a low  $d$  factor is more appropriate for scripts that belong to more than one domains.

### 5.2 Content Summarization

In order to achieve summarization we need to specify another parameter called hit ratio or  $hr$ . In order to compute  $hr$ , we define a sentence as the set of words between the beginning of a paragraph and a dot, between two dots, between a dot and a question-mark and vice versa, between a semicolon and a dot and vice versa and between an exclamation mark and a dot and vice versa. Also, all bulleted or numbered sentences as well as the starting sentence, that ends in “:” will be treated as one sentence. Let  $f$  be the number of nouns of a sentence and  $D$  the number of words of the domain the document belongs to.  $hr$  is computed as:

$$hr = f \cap D / f$$

Again  $f$  is computed using the stemming algorithm proposed in Chapter 3. If the document belongs to more than one domain then  $hr$  is computed per domain and the total hit ratio is assumed as the summation of the consequent hit ratios. In a hash table, we store an index of the position of a sentence on a text and the  $hr$  factor. We then construct the summary of the text sorting the results. The summary is made up of the sentences with the highest hit ratio maintaining the order of appearance in the text. The bigger the summary the user requests, the less important sentences according to  $hr$  variable, would be included in the summary. In cases of sentences scoring an equal  $hr$  we prefer to include the larger possible one.

The algorithm “breaks the rule” of using only nouns in content summarization by considering particular modal verbs expressing severely negative or positive status of sentences as «δεν μπορεί» (cannot), «δεν πρέπει» (must not), «πρέπει» (must). These words, when included in a sentence, depict meanings that have greater possibility of being included in the final summary, due to certainty or uncertainty respectively.

Other words that must be treated in the same way are words as «θάνατος» (death), «ειρήνη» (peace), «διαφθορά» (corruption) and generally words that stand for meanings that are either strongly positive or strongly negative.

A feature that is still under consideration is the possibility of applying a weight label on the keywords of a domain. In “politics” for example, “elections” would have a greater weight label than “resources”, because “elections” when used in a text, has greater possibility of illustrating that the thematic subject of the text is “politics” than “resources”.

The methodologies described above correspond to the following algorithm:

*Doc: the document to summarize*

$MScript\{1..p\}$  set of sentences of the document ( $Doc = MScript_1 \cup MScript_2 \dots \cup MScript_p$ )

$|MScript_p|$  is the number of words (all kind of words, verbs, nouns, adjectives....) of the  $p$ -th sentence

$HTable(SentenceIndex, HitRatio)$  Hash table including the sentence ( $Mscript$ )

$SentenceIndex \in \{1..p\}$  and the  $HitRatio$  of the sentence

$BD \{D1..Dr\}$  set of domains that the doc belongs to

$D_i\{1..n\}$  set of nouns of a  $i$ -th semantic domain

$V\{1..k\}$  set of nouns of  $Doc$

$F_j\{1..m\}$  set of nouns of the  $j$ -th sentence  $MScript_j$  in the document ( $V = F_1 \cup F_2 \dots \cup F_p$ )

$\alpha$  the number of words of the summary we intent to produce

$Summary(j)$  a table of the sentences that will produce the summary

“Step 1. Text Classification”

Begin

For  $i=1$  to  $n$

```

 $d(i) = (|V| \cap |Di|) / |V|$ 
if  $d(i) > 0.3$  then
   $BD\{\} = BD\{\} \cup \{i\}$  "Doc belongs to  $Di$  semantic domain"
end_if
end_for

```

*"Step 2. Content Summarization"*

```

For each  $Di \in BD$ 
  For  $j=1$  to  $p$  "p is the number of sentences"
     $hrj = hrj + |fj \cap Di| / |fj|$ 
  HTable( $j, hrj$ )
end_for
end_for each

```

*Sort HTable ASC hrj "Sort HTable in Ascending order of hrj"*

```

HTable_index = 1
While (no_of_words_in_summary <  $\alpha$ ) and (HTable_index <=  $p$ )
   $j = \text{HTable}(\text{HTable\_Index}, 1)$  "content of HTable in row HTable_Index and column 1"
  Summary( $j$ ) = MScript $k_j$ 
  no_of_words_in_summary = no_of_words_in_summary + |MScript $j$ |
  HTable_Index = HTable_Index + 1
End_While

```

```

For  $j=1$  to  $p$ 
  Final_Summary = Final_Summary + Summary( $j$ )
End_for

```

*End*

## 6. AN EXAMPLE OF THE PROPOSED ALGORITHM

Let us provide an example of the algorithm on a news article, taken from news site <http://www.pathfinder.gr>. For clarity of the presentation, we will demonstrate the application of the second step of the algorithm, taken that the domain is already defined for this document, and that the words setting up this domain are known. The domain of the specified news article is "accidents". The words setting up this domain  $D$  are typically, and by approximation:

$D$ : {θάνατος, καταστροφή, χιονόπτωση, χιονοθύελλα, κατολίσθηση, τυφώνας, κυκλώνας, τάφος, ζωή, επιζών, νοσοκομείο, συνθήκες, διάσωση, συνεργείο, τραυματίας, πεθαμένος, γιατρός, πυροσβέστης, ξηρασία, χαλάζι, τραγωδία,

λεωφορείο, αυτοκίνητο, τρένο, αεροπλάνο, πλοίο, φορτηγό, επιβάτης, οδηγός}

In English the set would be:

D: {death, catastrophe, snowfall, blizzard, snowstorm, landslide, typhoon, cyclone, hurricane, tomb, life, alive, survivor, hospital, conditions, rescue, party, wounded/injured, dead, doctor, fireman, drought, hail, tragedy, bus, car, train, airplane, ship, truck, passenger, driver}

The news article to be summarized is the following:

#### **“Τραγωδία με λεωφορείο στην Ελβετία**

12 νεκροί και 15 τραυματίες, εκ των οποίων οι 4 σοβαρά από ελεύθερη πτώση λεωφορείου σε χαράδρα

12 άνθρωποι έχασαν τη ζωή τους από πτώση λεωφορείου με 27 επιβάτες, σε χαράδρα στο καντόνι Βαλέ της Ελβετίας, κοντά στα σύνορα με την Ιταλία. Το δυστύχημα σημειώθηκε γύρω στις 8, ώρα Ελλάδος, όταν λίγο μετά το παραμεθόριο χωριό Ορσιέρ, ο οδηγός του λεωφορείου έχασε τον έλεγχο, καθώς ο δρόμος ήταν ολισθηρός λόγω των χιονοπτώσεων. Στο τέλος της "τρελής" διαδρομής, το λεωφορείο κατέπεσε στο βάθος χαράδρας 150 μέτρων. Άλλα σώματα επιβατών εκσφενδονίστηκαν, άλλα παγιδεύτηκαν στο εσωτερικό του λεωφορείου, το οποίο κατέληξε στον πυθμένα του ποταμού που διαρρέει την χαράδρα. Για τη διάσωση των επιζώντων παίρνουν μέρος περίπου 200 μέλη σωστικών συνεργείων - γιατροί, αστυνομικοί, πυροσβέστες ακόμα και συνοροφύλακες- ωστόσο το έργο τους είναι εξαιρετικά δύσκολο καθώς στην περιοχή επικρατούν άσχημες καιρικές συνθήκες. Σύμφωνα με σωστικά συνεργεία, 15 είναι οι τραυματίες, εκ των οποίων οι τέσσερις πολύ σοβαρά. Το λεωφορείο είχε ξεκινήσει από προάστιο της Βέρνης, με τελικό προορισμό το ιταλικό λιμάνι της Σαβόνα, όπου οι επιβάτες (24 Ελβετοί τουρίστες κι η ξεναγός τους) θα επιβιβάζονταν σε κρουαζιερόπλοιο. Στο λεωφορείο υπήρχε και "αναπληρωματικός" οδηγός.”

A translation in English of this is the following:

#### **“Bus tragedy in Switzerland**

12 people died and 15 were wounded, 4 of whom seriously, in a free fall of a bus in a gorge.

12 people lost their lives from a bus fall in a gorge of Valle canton in Switzerland near its borders, with Italy. The accident occurred around 8 am



Greece time, when just after village Orsier, the bus driver lost control of the bus, due to slippery road and snowfall. The bus ended up in a gorge 150 m. deep. Some of the passengers were dashed out of the bus, while some others ended up on the bottom of the river that runs through the gorge. For the rescue of the passengers, about 200 members of rescue parties participate – doctors, police officers, firemen and border guards-but their task is too hard to accomplish, due to bad weather conditions in the area. According to the rescue parties, 15 people are injured, 4 of whom are in very serious condition. The bus had departed from a suburb of Bern, with a destination of Italy's port of Savona, where the passengers (24 Swiss tourists and their guide) would board on the cruise ship. There was a replacement driver on the bus."

The algorithm first defines the number of sentences and indexes them in a set. The set of sentences is as follows:

(Notation: in the following representation, the set is structured as [sentence, number of nouns belonging to the domain present on the sentence, total number of nouns])

Doc={[(Δώδεκα νεκροί ... σε χαράδρα),(3),(5)],[(Δώδεκα άνθρωποι...με την Ιταλία),(3),(11)],[(Το δυστύχημα ... των χιονοπτώσεων),(3),(8)],[( Στο τέλος ... 150 μέτρων),(1),(5)],[(Άλλα σώματα ... την χαράδρα),(2),(7)],[(Για την διάσωση ... άσχημες καιρικές συνθήκες),(6),(11)],[(Σύμφωνα με τα σωστικά...πολύ σοβαρά),(2),(2)],[(Το λεωφορείο...κρουαζιερόπλοιο),(2),(10)],[(Στο λεωφορείο...οδηγός),(2),(2)]}

Then, it searches the words of the specified domain inside Doc set. Simultaneously, it creates the hashing table by computing the hr factor. The resulting hashing table is:

Sentence	hr factor
Δώδεκα νεκροί ... σε χαράδρα	0.6
Δώδεκα άνθρωποι...με την Ιταλία	0.27
Το δυστύχημα ... των χιονοπτώσεων	0.375
Στο τέλος ... 150 μέτρων	0.2
Άλλα σώματα ... την χαράδρα	0.286
Για την διάσωση ... άσχημες καιρικές συνθήκες	0.545
Σύμφωνα με τα σωστικά ... πολύ σοβαρά	1

Το λεωφορείο...κρουαζιερόπλοιο	0.2
Στο λεωφορείο...οδηγός	1

Table 1. Hashing table of the document

The next step is to define the number of words we want to include in the summary. Let  $\alpha$  be the number of words and for this case equals to 20. Then, by the hashing table, the Doc set and the relative position of sentences in the text the resulting summary is:

**“Τραγωδία με λεωφορείο στην Ελβετία**

Σύμφωνα με σωστικά συνεργεία, 15 είναι οι τραυματίες, εκ των οποίων οι τέσσερις πολύ σοβαρά.”

In English the resulting summary is:

**“Bus tragedy in Switzerland**

According to the rescue parties, 15 people are injured, 4 of whom are in very serious condition.”

If  $\alpha$  becomes 30 then the extracted summary is:

**“Τραγωδία με λεωφορείο στην Ελβετία**

Σύμφωνα με σωστικά συνεργεία, 15 είναι οι τραυματίες, εκ των οποίων οι τέσσερις πολύ σοβαρά. Στο λεωφορείο υπήρχε και "αναπληρωματικός" οδηγός.”

In English the resulting summary is:

**“Bus tragedy in Switzerland**

According to the rescue parties, 15 people are injured, 4 of whom are in very serious condition. There was a replacement driver on the bus.”

If  $\alpha$  in turn becomes 40 then the summary is:

**“Τραγωδία με λεωφορείο στην Ελβετία**

Δώδεκα νεκροί και 15 τραυματίες, εκ των οποίων οι τέσσερις σοβαρά από ελεύθερη πτώση λεωφορείου σε χαράδρα. Σύμφωνα με σωστικά συνεργεία, 15 είναι οι τραυματίες, εκ των οποίων οι τέσσερις πολύ σοβαρά. Στο λεωφορείο υπήρχε και "αναπληρωματικός" οδηγός.”

In English the resulting summary would be:

#### **“Bus tragedy in Switzerland**

12 people died and 15 were wounded, 4 of whom seriously, in a free fall of a bus in a gorge. According to the rescue parties, 15 people are injured, 4 of whom are in very serious condition. There was a replacement driver on the bus.”

If  $\alpha$  becomes 80 or if we decide to include another sentence in the summary then the sentence to be included is the one with  $hr=0.545$ . Its positioning the text is relative to the original document. Therefore, the resulting summary is:

#### **“Τραγωδία με λεωφορείο στην Ελβετία**

Δώδεκα νεκροί και 15 τραυματίες, εκ των οποίων οι τέσσερις σοβαρά από ελεύθερη πτώση λεωφορείου σε χαράδρα. Για τη διάσωση των επιζώντων παίρνουν μέρος περίπου 200 μέλη σωστικών συνεργείων -γιατροί, αστυνομικοί, πυροσβέστες ακόμα και συνοροφύλακες- ωστόσο το έργο τους είναι εξαιρετικά δύσκολο καθώς στην περιοχή επικρατούν άσχημες καιρικές συνθήκες. Σύμφωνα με σωστικά συνεργεία, 15 είναι οι τραυματίες, εκ των οποίων οι τέσσερις πολύ σοβαρά. Στο λεωφορείο υπήρχε και "αναπληρωματικός" οδηγός.”

The English version of this summary is:

#### **“Bus tragedy in Switzerland**

12 people died and 15 were wounded, 4 of whom seriously, in a free fall of a bus in a gorge. For the rescue of the passengers, about 200 members of rescue parties participate – doctors, police officers, firemen and border guards-but their task is too hard to accomplish, due to bad weather conditions in the area. According to the rescue parties, 15 people are injured, 4 of whom are in very serious condition. There was a replacement driver on the bus.”

## **8. CONCLUSIONS**

In this paper, we present basic principles and an algorithm for extracting document content summary for Greek language, by using statistic methods. The algorithm presented is based on grammatical rules and semantic information dedicated for Greek language. The algorithm has been tested on a variety of news articles and produces satisfactory results for a variety of thematic subjects. Apart from the algorithm, a test case is presented to validate its performance.

## REFERENCES

- Baker D. and McCallum A., 1998, Distributional Clustering of Words for Text Classification, In Proceedings of ACM SIGIR 98, Melbourne, Australia.
- Barzilay, Regina, and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, 10–17. Madrid, Spain.
- Baxendale, P.B. 1958. Machine-made Index for Technical Literature - an experiment. IBM J. Res. Dev. 2(4): 354–361.
- Benbrahim, M., and K. Ahmad. 1995. Text Summarisation: the Role of Lexical Cohesion Analysis. The New Review of Document & Text Management 321–335.
- Charras C., Lecroq T., 2004, Handbook of Exact String-Matching Algorithms, London King's College.
- Edmundson, H.P. 1969. New Methods in Automatic Extracting. Journal of the Association for Computing Machinery 16(2): 264–285.
- Leskovec J., Grobelnik M., Milic-Frayling N., June 2004, Learning Semantic Graph Mapping for Document Summarization, Solomon Seminar, Jozef Stefan Institute, Slovenia.
- Luhn, H.P. 1958. The Automatic Creation of Literature Abstracts. IBM Journal of Research Development 2(2): 159–165.
- Porter, M.F., 1980, An algorithm for suffix stripping, Program, 14(3) :130-137
- Teufel, Simone, and Moens Moens. 1999. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In I. Mani and M.T. Maybury, eds., Advances in Automatic Text Summarization, 155–171. The MIT Press.