ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 7, No. 1, 2012, *pp*. 025-036

Document Classification in Summarization

Georgios Mamakis[1], Athanasios G. Malamos[2], J. Andrew Ware[3], Ioanna Karelli[4]

[1]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

[2]Department of Applied Informatics and Multimedia. Technological Educational Institute of Crete, Heraklion Crete, Greece, Email: amalamos@epp.teicrete.gr

[3]Faculty of Advanced Technology, University of Glamorgan, Trefforest, Wales, Email: jaware@glam.ac.uk

[4]Faculty of Philology, University of Crete, Rethymnon, Crete, Email: iwannakarelli@gmail.com

(*Received August 13, 2011, accepted September 15, 2011*)

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 supervises and unsupervised approaches and yields significant results.

Keywords: Document classification, supervised document summarization, statistics

1. Introduction

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 extend 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 uni-gram 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

DOI https://doi.org/2024-JICS-22660 | Generated on 2025-04-20 17:40:36

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) indentify 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

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

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

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

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

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

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 (4). 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

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

where 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

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

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

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

where α 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 Kalamboukis [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 (μεν). 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. Te stemmer's performance may be affected in cases where a noun can be used as an adverb (e.g. αλήθεια- 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 ad 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.

Table 1. Class profiles

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

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

Table 2. Overall Classification Results

		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

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

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

where word denotes the occurrences of term word in a random document, Σword the total number of words in the document, traindoc the total number of documents in the training set and traindoc,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

$$A = U \times \Sigma \times V^T \tag{8}$$

where U is a column-orthogonal matrix holding the left singular vectors, Σ 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 Σ 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 Σ is a diagonal matrix whose values are ordered descending, then it is safe to consider that the top r topics from Σ 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

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

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:

Table 3. Summarization Results

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

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 α

= 1.2 in formula (6), the results acquired were:

Table 4. Summarization Results with positional characteristics

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

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 se 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*. 2000, **1**: 495-501. doi:10.3115/990820.990892.

[2] M. F. Moens, R. Angheluta and J. Dumortier. Generic technologies for single- and multi-document sumarization. *Journal of Information Processing and Management*. Elsevier, 2005, **41**: 569-586.

[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 18th International Conference on Research in Information Retrieval (SIGIR ' 95).* 1995, pp. 55-60.

[5] C. Aone, M. Okurowski, 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*. 1997, pp. 66-73.

[6] I. Rish. An empirical study of the Naive Bayes Classifier. *in Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*. 2001, pp. 41-46.

[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*. AAAI Press. 1998, pp. 41-48.

[8] K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. in Machine Learning. *Special Issue on Information Retrieval*. Kluwer Academic Publishers, ISSN:0885-6125. **23**(2-3): 103-134.

[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*. AAAI Press. 2007, pp. 540-545.

[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*. SAGE publications. June 2011, **37**(3): 293-303.

[12] M. Srikanth, R. Srihari. Biterm language models for document retrieval. *in Proc. of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval.* Tampere, Finland. 2002, pp. 425-426.

[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.* Kluwer Academic Publishers, ISSN:1573-7659. 2004, 7(3-4): 317-345.

[15] W. Cheng, E. Hullermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*. doi:10.1007/s10994-009-5127-5. 2009, **76**: 211–225.

[16] Zhang, M.-L., & Zhou, Z.-H. ML-kNN: A lazy learning approach to multilabel learning. *Pattern Recognition*. 2007, **40**(7): 2038–2048.

[17] Godbole, S. & Sarawagi, S.. Discriminative Methods for Multi-labeled Classification. *Lecture Notes in Computer Science*. DOI: 10.1007/978-3-540-24775-3_5. 2004, **3056/2004**: 22-30.

[18] H.P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal*. 1958, pp.159-165.

[19] P.B. Baxendale. Machine-Made Index for Technical Literature – An experiment. *IBM Journal*. 1958, pp. 354-361.

[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*. Prague, Czech Republic, 2007, pp. 448-457.

[22] M. Fattah, and F. Ren. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech and Language.* 2009, **23**: 126-144.

[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.* ISBN 0-7803-9270-1, Cairo, Egypt. 2005, pp. 579-591.

[24] C. Bouras, V. Tsogkas. Improving Text Summarization Using Noun Retrieval Techniques. in Lecture Notes in Computer Science, Springer Berlin / Heidelberg. 2010, **5178**: 593-600.

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

[26] T. Z. Kalamboukis. Suffix stripping with modern greek. in *Program: electronic library and information systems.* 1995, **29**(3): 313 – 321.

[27] S. Scott, S. Matwin . Feature Engineering for Text Classification. *in Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kauffman Publishers, ISBN:1-55860-612-2. 1999, pp. 379-388.

[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*. doi:10.1108/eb026526. 1972, **28** (1): 11-21.

[33] Y. Gong and X. Liu. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. *Proc. SIGIR'01.* September 2001, pp. 19-25.

[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*. doi:10.1145/1361684.1361686. 2008, **26**(3): 1–37.

[36] G. Erkan, D. R. Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial In telligence Research*. 2004, **22**: 457-479.

[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

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

[40] A. Nenkova. Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference. *20th National Conference on Artificial Intelligence (AAAI 2005)*. Pittrsburg.