Optimization in Extractive Summarization Processes through Automatic Classification

Angel Luis Garrido[1], Carlos Bobed[1 2], Oscar Cardiel[3], Andrea Aleyxendri[3], and Ruben Quilez[3]

[1] Dept. of Computer Science and Systems Engineering, University of Zaragoza, Spain
[2] Aragon Institute of Engineering Research (I3A), Spain
{garrido,cbobed}@unizar.es
[3] IT Department, Grupo Heraldo,
Zaragoza, Spain
{ocardiel,aaleyxendri,rquilez}@heraldo.es

Abstract. The results of an extractive automatic summarization task depends to a great extend on the nature of the processed texts (e.g., news, medicine, or literature). In fact, general-purpose methods usually need to be adhoc modified to improve their performance when dealing with a particular application context. However, this customization requires a lot of effort from domain experts and application developers, which makes it not always possible nor appropriate.
In this paper, we propose a multi-language approach to extractive summarization which adapts itself to different text domains in order to improve its performance. In a training step, our approach leverages the features of the text documents in order to classify them by using machine learning techniques. Then, once the text typology of each text is identified, it tunes the different parameters of the extraction mechanism solving an optimization problem for each of the text document classes. This classifier along with the learned optimizations associated with each document class allows our system to adapt to each of the input texts automatically. The proposed method has been applied in a real environment of a media company with promising results.

Keywords: Extractive Summarization, Optimization, Machine Learning, Automatic Classification

1 Introduction

Automatic text summarization consists of decreasing the size of a given text while retaining its most relevant information. It is a challenge task that requires an extensive knowledge of the context of the text, its structure, and its writing style. Automatic text summarization is increasingly used both in research and industry, in areas such as information retrieval [1], question answering [2], or data mining [3]. Besides, the existence of the World Wide Web has caused an explosion in the amount of textual information in all areas, so automatic summarization becomes a great tool for accessing information in a consistent and summarized way.

These techniques are divided into two categories: *extractive summarization*, and *abstract summarization*. The former ones are produced by concatenating several sentences literally, while the latter ones require transformation of the sentences by deleting, substituting, and rearranging them. Summarization algorithms based on extractive techniques are generally simpler to implement, and they are mainly based on statistical or linguistic approaches to find the most relevant sentences to be included in the final summary [4]. This approach allows a simple way of working with any kind of text in any language, regardless of the textual context. In recent years new approaches have been studied in order to improve the outcomes: detection of co-ocurrence [5], simplification of sentences [6], use of named entities [7], etc. However, one of the problems with such algorithms is that a general approach prevents optimal results when it is applied in a particular domain, which leads us to the importance of the application context [8].

Hence, when an extractive summarization algorithm is used in a professional context, it is very common to perform tasks that improve the efficiency of the algorithm according to the context in which it works. For example, in the news context, it may be very interesting to take advantage of the existence of a title, a subtitle, or even the caption of the photographs accompanying the news, in order to modify the sentence selection procedure for generating the final summary. Of course, this method can not be used if what we are doing is to summarize another type of text, such as a judicial sentence, which lacks all of these elements. Again in the news domain, we can find that it is not the same to summarize a teletype, or a musical review, or an interview. All of these formats can have large structural differences that reduce the performance of generalist algorithms, but such characteristics can be exploited by customizable expert systems.

In this paper, we propose a supervised learning methodology to automatic extractive summarization, which is based on leveraging the features of the documents in order to classify them *before* the summarizing process. Firstly, we train a model to classify the documents we are working with. Then, we solve an optimization problem for each of the text document classes to learn the optimum parameters that guide the extraction of sentences in each of the cases. The combination of both methods in the system allows us to achieve better results.

The main contribution of this work is to improve the generation process of a extractive summary from a single-source document with a multi-language and general focus. That is achieved by combining an automatic categorization of texts with performing a personalized adjustment of the summarization process according to this categorization. We have implemented this methodology in the development of a system devoted to summarizing news according to their typology. We have tested that system using a real dataset, which we make available for other researchers. The experiments performed show the multilingual capabilities of our approach, as well as a good outcome on a real working environment.

This paper is organized as follows: Section 2 studies the state of the art related to summarization close to this context. Our methodology is detailed in Section 3, and its application to a working system is presented in Section 4. Section 5 explains the different experiments we have performed and interprets the outcomes. Finally, Section 6 summarizes the key points of this work, provides conclusions, and explores future work.

2 Related Work

Interest in automatic summaries appeared back in the 50's. Luhn suggested in [9] to weight the sentences of a document as a function of high frequency words and disregarding the very high frequency common words. Apart from such approach, other methods valued the use of certain words (*cue words*), the headers of the document, and the structure [10] in order to determine the weight of each sentence.

In the 1990s, machine learning techniques started to be used in Natural Language Processing (NLP) and, therefore, also in text summarization. At the beginning, most systems relied on naive-Bayes methods [11], but others focused on learning algorithms that make no independence assumptions [12]. More recently, some works have used hidden Markov models [13], log-linear models [14], and even neural networks [15] to improve extractive summarization.

Some recent works have leveraged the context regarding summarization. For example, in [16] the authors suggest and bring experimental evidence about that the effectiveness of sentence scoring methods for automatic extractive text summarization algorithms depends on certain features of each document typology, working with news, blogs, and articles. Another inspiring work is [17], where authors face the problem of the Twitter context summarization by adapting certain environment signals in the context of the tweet. Both approaches are interesting but lack the generality necessary to be applied in very different contexts.

As seen, there are a variety of approaches for generic summarization applicable when the purpose of the reader is unknown. But the main drawback of a generic summarization is the difficulty of getting precise results. However, this performance is a strong requirement in real environments. Hence, new proposals with the aim of covering this need are required.

3 Methodology

This section describes a working methodology applicable to any system dedicated to produce extractive summaries from single-source text documents.

Our method customizes and enhances the process for each existing document typology. The key for improving is to identify the typology of the source documents, and then, to automatically obtain the most suitable parameters of the summarization process with the aim of improving the outcomes for each type. This is done in an off-line step over a corpus which represents the documents the system is going to process in the production stage. Then, in the summarization stage, the system's input is a text that will go through a set of specialized treatments until the generation of the output, the final summary. The size of that summary is established by the *compression rate (CR)*, given by a specific number of words.

In the following, we focus on explaining the training stage of our methodology. Firstly, we introduce the definitions of the different elements in the problem. Then, we move onto the details of both training steps, the automatic classification of documents, and the optimization of the summarization parameters.

3.1 Definitions

Before starting, it is important to give a formal definition of the different elements that take part in the problem. In this paper, we consider that a document *D* is defined as a tuple $< T, A >$, where:

- T is the text to be summarized. We consider it as being formed by the ordered set $S = \{s_1, \ldots, s_n\}$, with s_i being each of the sentences in order of appearance in the text.
- A is a set $\{a_1, \ldots, a_n\}$, with a_i being attributes of the text T which can be interesting for the elaboration of its summary.

Each of these attributes a_i is formed by a tuple $< name_i, \{value_i\} >$, as well. Examples of attributes could be: title, subtitle, author, place, etc. Note how an attribute can be multivalued in our setting; for example, several captions may correspond to a same text *T*.

Thus, given a document D, our goal is to obtain a set $R \subset S$, which contains the most relevant sentences, keeping their order. To assess the relevance of a particular sentence within such document, we need a function $ValF$, such as:

$$ValF : String \times \{D\} \rightarrow \mathcal{R}^+$$

with *String* being the set of all possible text strings, and $\{D\}$ the set of all possible documents. Besides, as above mentioned, we have to bear in mind the size constraint imposed by the *compression rate* (CR), which can be given by a specific number of words.

So, in general, given a document D a summarization system would return a set of sentences R such as:

$$R = \{r_i | r_i \in S \ \wedge \ \nexists s_j \in S . ValF(s_j, D) > ValF(r_i)\}$$

satisfying

$$\sum_{i=1..|R|} size(r_i) < CR$$

with S being the set of sentences in D, and *size* a function that gives the word count of a given sentence. It can be considered that, given a set of texts, the optimal way to make an extractive summary of each is to devise a particular *ValF* function which provides an optimal result over the corpus. The evaluation of the result is usually be done by comparing each of the summaries obtained with a set of model summaries, for which a pre-established comparison function *(CompF)* is used.

However, this *ValF* function might be optimal for the corpus globally, but not for each of the different text typologies that might be included in the corpus. Thus, we advocate for finding first a categorization of the different texts and styles that might be the target of our system, and, for each of the categories, obtain an optimized *ValF* function in an automated way.

3.2 Automatic Classification of Input Documents

Many works can be found focused on finding an optimal function *ValF* in order to obtain optimal results. In this work, as above mentioned, we propose an additional consideration that improves the application of these techniques to real-world scenarios. Our proposal is that, given a set of texts, *there may be a set of ValF functions that provide an optimal result, corresponding to the different typologies of texts existing in the set*. That is, if we are able to apply a specific *valF* function on each type of text, we will achieve better results with the application of a generic *ValF* function.

Therefore, we consider obtaining the set of text typologies within the set of documents as the starting part of the methodology. We denote the set of such typologies as $P = \{p_1, \ldots, p_x\}$. For performing this categorization, both the text *T* of each document and its set of attributes *A* can be taken into account. Note that the typologies can be established *a priori* by domain experts adopting a supervised approach, or they could be obtained by applying an unsupervised clustering algorithm (e.g., K-clustering, or hierarchical clustering algorithms). The particular technique to detect the different underlying text typologies is out of the scope of this work.

Although categorization of each input document could be manually performed, it would be very expensive and time-consuming. Therefore, once the categories have been defined and assuming that there is an acceptable set of hand-made examples of summaries for each category, we advocate for using an automatic classification using a supervised machine learning method, for example naive bayes, support vector machines, random forests, or artificial neural networks. This input categorization guides which $valF$ is going to be used in the summarization process, adapting the system automatically to the input presented to it.

3.3 Summarization Process

The next step is to find out the most suitable *ValF* for each of document typology present in our system. Again, the collaboration of experts and custom development would be the most valuable method in order to design optimal *ValF* functions in real contexts for each of them.

Nevertheless, it is clear that this ideal situation is not always feasible. Therefore, it is necessary to find a method as automatic as possible. Nowadays, we can find a great deal of off-the-self generic extractive summarization approaches which work directly on the text of the document, so we can assume that we can always select one of the relevance evaluation functions they use as baseline, which we will call *BaseVal*. As we want to adapt its evaluation to the different typologies, we propose to extend it externally by adding different terms, so we can build a family of $ValF$ functions defined by:

$$ValF_{\alpha}^{\{\beta_i\}}(s, D) = \alpha * BaseVal(s, T) + \sum_{i=1..n} \beta_i * attrVal(s, D)$$

with s being the sentence to be evaluated, and $attrVal$ being functions that evaluate s according to different attributes of the document[4].

[4] In fact, they belong to $ValF$ family of functions as well, but for the sake's of readability we have decided to change their name.

In this way, we can automatize the adaptation of the extraction to the different typologies by tuning the α and $\{\beta_i\}$ parameters that weight the generic approach and the extensions over the attributes we use to extend the function[5]. As it happened with the classification processes, we can also assume that at least an acceptable set of model summaries are available for each of the document classes. For each of the defined categories, the summaries belonging to them will be used to optimize and adapt the weight values, and, doing so, to obtain a particular $ValF$ function for such document typology.

In our work, we have devised and applied some methods we have found useful for improving the performance of different summarization processes. In particular, we suggest:

1. *Exploring Attributes*: Firstly, for each attribute of a document D, the substantives in the attribute value are obtained. A relevant presence of these words in the training summaries may indicate that the sentences containing such words should be more relevant.
2. *Vocabulary Analysis*: The frequencies of the words used in the summaries are obtained. After studying them, we have found that that there is a correlation among the types and quantities of words of each type, and the type and style of the text to be summarized. So, we gradually increase the relevance of the sentence containing them in each type of document summarization.
3. *Sentence order*: First, the order of the sentences in the examples summaries is checked. If a significant percentage of sentences are located in a certain area of the documents, we leverage that circumstance by increasing the scores of these sentences gradually.

If the three methods are combined along with a selected *baseline* function, solving a multiple optimization problem [18] using separately each of the subsets of documents of each document typology give us the values for the parameters to maximize the result in each situation. Once we have calculated all the parameter values for each of the document typologies, we just have to classify the documents using the trained text classifier, and then apply the optimized $ValF$ for such typology.

4 Applying the Methodology

In order to validate our methodology, we have applied it to a summarization system on a real-world application. In particular, we have chosen the context of the news produced by a media company, as it is appropriate to evaluate the performance of a system of these characteristics. We have designed a system called NESSY (NEws Summarizer SYstem), a news extraction-based summarization system customized for the specific treatment of news, interviews, briefs, editorials, letter to the editors, and reviews. The system is devoted to perform the task of creating a single summary from a single-source document customizing the process for each typology.

In this section, firstly, the theoretical context of the news text is introduced. Secondly, the application of the methodology to develop the system is detailed.

[5] We are aware we could get rid of the baseline term, but it is useful for the sake of comparing our approach with generic approaches.

4.1 News Genres and Structure

The purpose of news is to report on events and topics of general interest. Journalistic genres are ways of written communication that differ according to the needs or objectives of who write it. Overall, experts generally agree that there are three main journalistic genres: Informational, Interpretative, and Borderline [19–21].

1. *Informational*: They aim to narrate the news with an objective and direct language. The person writing the text does not appear explicitly. The texts are informative when transmitting data and facts of interest to the public, whether new or known in advance. The information does not allow personal opinions, much less judgmental.
2. *Interpretative*: They are intended to express the point of view of who writes. The author interprets and discusses reality, evaluates the circumstances in which the incident occurred, and he/she expresses judgments on the reasons and the consequences that may arise from them.
3. *Borderline*: Those in which, in addition to report an occurrence or event, the journalist expresses his opinion. Its purpose is to relate the event to the temporal and spatial context in which it occurs.

These genres are further subdivided into different types, which can be appreciated in Table 1. Each typology has its own characteristics that identify it: the text structure, linguistic aspects, use of verbal forms, explanation of technical terms, syntax, quotes, signatures, and the use of rhetorical figures, are examples of features.

Table 1: Types of news for each of the journalistic genres.

Informational	**Interpretative**	**Borderline**
Report	Review	Chronicle
Journalistic report	Editorial	Opinion
Biography	Newspaper column	Letter to the editors
Press review	Obituary / Farewell	Journalistic interview
Interview	Journalistic article	Debate
Documentary report		Talk show
Brief news		

Furthermore, news presents a number of common structural elements [22]:

- *Headline*: Short set of sentences, including the most relevant information, which intend to capture the attention of readers.
- *Caption*: One or few sentences located below the photos or pictures accompanying text.
- *Body*: Set of paragraphs that make up the text and details on the topic. It is the longest section, and its structure is an inverted pyramid regarding the importance of information, whose top is the first paragraph (called *lead*). The lead is situated at the beginning of the body in order to catch the reader's attention.

– *Layout Resources*: Mass media use strategies and resources to capture the reader's attention, which provide extra information or highlight the most important aspects. Examples can be: quotes, documentary data, tables, etc.

Summaries of each type of news must leverage different attributes in different ways in order to achieve good results. It is mainly for this reason that the context of the news has been chosen as an appropriate use case for the evaluation of our methodology.

4.2 The NESSY System

The aim of this system is the individual summarization a piece of news (document) which belong to one of the types of news journalistic genre previously mentioned (see Section 4.1). Each document type has unique features that can be used for classification purposes: writing style, structure, presentation, predominant vocabulary, linguistic resources, etc.

Following our proposed methodology, the system is divided into two steps: Text Classification, and Text Summarization. They are directly the application of the steps of our proposal.

Text Classification First, the system categorizes the input news by using support vector machines (SVM), a well known supervising learning model [23]. The reasons for choosing this methodology are several [24]: 1) SVMs are able of extracting an optimal solution with a very small training set size; 2) as SVMs uses the feature space images by the kernel function, SVMs are applicable in such circumstances that have proved difficult or impossible for other methodologies like bayes or back-propagation neural networks (for example, when data is randomly scattered, and when the density of the distribution of the data is not even well defined); and 3) last but not least, for its simplicity and its speed. In addition, our previous experience with this tool has demonstrated several times [25–28] its good performance in this type of scenarios.

The correct categorization of the input document determines the following stage. In NESSY, the following types of news have been considered: analysis, editorial, interview, letter, opinion, piece of news, report, review, short piece of news, and documentation.

Text Summarization This second step is in charge of applying the actual summarization process. As stated before, the system has been tuned offline by optimization techniques with the objective of applying the most suitable *ValF* function to each of the categories.

If we analyze the different categories from the point of view of the methodology proposed in Section 3, we find that the summaries of each category really can be optimized if the system leveraging the relevant features from each of them. In particular, we present here some relevant examples that have been considered in NESSY successfully:

- *Standard Report:* Journalistic texts of this type are clear and concise, and consist of a recent event that has an interest or curiosity for readers. The title, the caption, and the first paragraph are elements that usually contain the most relevant words, and therefore sentences with these words most likely should appear in the summary.
- *Interview:* An interview is composed by a series of questions, and their answers. The percentage of question marks and interrogative words (*who, when, where*, etc.) compared with the total amount of words is frequently higher than other types of news. The interviewee's name and his/her main quote is usually found in the headline, along (typically) with his/her profession. If an introduction paragraph exists, it usually contains relevant keywords and named entities. The first questions and the included ones into the Layout Resources are also typically the most important.
- *Review:* In this kind of text, in which the writer tries to explain his/her opinion about artistic productions, the main characteristic is that vocabulary is quite repetitive. In particular, if there exists any caption in, it is usually very important.
- *Brief news:* The brief news are a set of short texts that are characterized by their brevity and conciseness. They are summarized news which kept only the most relevant data. The title and the very first sentence are the most relevant elements.

The summaries corresponding to the different categories can be improved if these types of modifications are taken into account when establishing the weights of the sentences to be extracted. We capture these different aspects in NESSY thanks to the use of the proposed *extensions* (*attrVal*) to improve the final *ValF* functions.

Once the text has been classified, NESSY only has to apply the appropriate *ValF* function with the off-line calculated *attrVal* functions, and retrieve the most relevant sentences according to the definition presented in Section 3.1.

5 Evaluation

To evaluate our approach, we have performed two different sets of experiments aimed at evaluating the performance of the complete setting, considering the quality of the resulting summaries. We first present the experimental settings and the datasets we have used, and then we detail and discuss the results for each of the experiments.

5.1 Experimental Settings

Datasets In our experiments, we have used two datasets:

- *DSHA-1* is a corpus composed by 14,000 news taken from *Heraldo de Aragón*[6], a major Spanish media. These news were previously categorized by the Documentation Department of the company with one of the next 10 types: Analysis, Editorial, Interview, Letter, Opinion, Piece of News, Report, Review, Shorts Piece of News, and Documentation. There are 1,400 news of each type. This dataset is used in both experiments for training and evaluating the text classifier precision, which is evaluated applying k-fold cross validation.

[6] http://www.heraldo.es

– *DSHA-2* is a smaller corpus of 400 news (40 of each aforementioned type) also taken from *Heraldo de Aragón*. Each of the news has an associated summary which has been made by professional documentalists. This dataset is used to test the precision and the recall of the summarization task.

Both datasets are available upon request to the authors exclusively for research purposes, subject to confidentiality agreements due to copyright issues.

Classifier Features As features for the classifier, the frequency of the each word in the text is used. To achieve the value of those features, stop words[7] are firstly removed from the text. Then, a lemmatization process is applied in order to extract the lemma[8] of each remaining word. The lemmatization process is usually useful on languages with declensions and a lot of verbal forms, such as French, German or Spanish, because it reduces the frequencies catalog. Finally, TF-IDF [29] algorithm is calculated for each lemma obtained from the text.

Results Comparison For comparison, we have used several on-line summarizers: SWESUM[9] (Sw), Tools4noobs[10] (T4n), Autosummarizer[11] (AS), and the Mashape Tools[12] (MT). We have configured all the summarizers to get a compression rate of 20%. This rate is easily translated to the required number of words, which is the CR we work with. Finally, we have used ROUGE-L to compare the automatic summaries obtained with the summary models in the dataset. ROUGE is a recall-based metric for fixed-length summaries which is based on n-gram co-occurrence, and ROUGE-L is one of the five evaluation metrics avaliable, and it is based on founding the longest common subsequence. It takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

5.2 Experiment 1

For this experiment 4,800 news from the dataset *DSHA-1* are selected, and 6 typologies are considered: Editorial, Interview, Letter, Piece of News, Review, and Short Piece of News. So, we have used 800 news of each type: 600 texts are used to train the model and 200 are used to test it. In this experiment different kernels and different types of multiclassifiers are employed. The techniques used are: SVM Multiclass with linear Kernel, SVM Multiclass with radial basis function (RBF) kernel, SVM Multiclass 4th degree polynomial kernel and SVM Binary Tree with RBF kernel. Table 2 shows the experimental results.

[7] Stop words are common words without relevant information (e.g. articles or conjunctions).
[8] A lemma is the canonical form of a word. For example, in English, sing, sings, sang, sung, and singing are different forms of the same verb, with "sing" as their common lemma.
[9] http://swesum.nada.kth.se/index-eng.html
[10] https://www.tools4noobs.com/summarize/
[11] http://autosummarizer.com/
[12] http://textsummarization.net/

Table 2: Accuracy results for 4-fold validation categorization test with 6 categories.

	Accuracy
SVM Multiclass linear kernel	87.45%
SVM Binary Tree RBF kernel	90%
SVM Multiclass RBF kernel	92.45%
SVM Binary Tree 4th degree polynomial kernel	92.59%

We selected the best classifier, in this case SVM Binary Tree 4th degree polynomial kernel (92.59% of accuracy), and we used it in the second task to sort a set of 240 news items (40 of each type mentioned before) from the *DSHA-2* dataset, corresponding to the six aforementioned types. After solving the multiple optimization problem we obtained six different *ValF* functions, which are used to customize the summarization process of each type of news. The results are shown in Table 3, where it can be seen how in those more specific categories the most significant improvements are achieved.

Table 3: F-measure results regarding a subset of the *DSHA-2* dataset, composed by 240 news, six types, and 40 news of each type. The ROUGE-L algorithm has been used to compare the summaries with the models.

	Sw	**T4n**	**AS**	**MT**	**Nessy**
Editorial	0.55	0.36	0.38	0.41	0.64
Interview	0.51	0.46	0.45	0.52	0.65
Letter	0.49	0.21	0.29	0.48	0.55
Piece of news	0.42	0.33	0.34	0.33	0.44
Review	0.36	0.37	0.33	0.26	0.46
Short Piece of News	0.51	0.46	0.45	0.52	0.65
Average	0.47	0.37	0.37	0.42	0.57

5.3 Experiment 2

For this experiment, the whole dataset (14,000 news) is used and the 10 categories are considered. In this case, 1,200 text of each genre are employed to train the model and 200 to test it. The techniques are the same as the used in the Experiment 1. Table 4 shows the experimental results.

We selected the best classifier, in this case SVM Binary Tree RBF kernel (83.59% of accuracy), and we used it in the second task to sort the complete dataset *DSHA-2*, composed of 400 news items (40 of each type mentioned before). After solving the multiple optimization problem we obtain ten different *ValF* functions, which are used to customize the summarization process of each type of news. The results are shown in Table 5.

Table 4: Accuracy results for 7-fold validation categorization test with 10 categories.

	Accuracy
SVM Multiclass RBF kernel	77.51%
SVM Multiclass linear kernel	77.73%
SVM Binary Tree RBF kernel	83.59%
SVM Binary Tree 4th degree polynomial kernel	37.90%

Table 5: F-measure results regarding the whole *DSHA-2* dataset, composed by 400 news, 10 types, and 40 news of each type using ROUGE-L for comparing.

	Sw	**T4n**	**AS**	**MT**	**Nessy**
Analysis	0.38	0.29	0.31	0.52	0.47
Documentation	0.31	0.24	0.27	0.28	0.39
Editorial	0.55	0.36	0.38	0.41	0.61
Interview	0.51	0.46	0.45	0.52	0.62
Letter	0.49	0.21	0.29	0.48	0.53
Opinion	0.41	0.35	0.36	0.39	0.51
Piece of news	0.42	0.33	0.34	0.33	0.44
Report	0.31	0.28	0.29	0.30	0.41
Review	0.36	0.37	0.33	0.26	0.45
Short Piece of News	0.51	0.46	0.45	0.52	0.62
Average	0.43	0.34	0.35	0.40	0.51

5.4 Discussion

As it can be seen in the previous tests, we have obtained satisfactory results, especially in Experiment 1 with more than 92% using the SVM Binary Tree with RBF kernel and the SVM Multiclass 4th degree polynomial kernel. However, when it has been included more categories the accuracy decreases to 83.59%. That is because it is difficult to distinguish between some categories with similar linguistics contexts, such as reporting, opinion or short. It is remarkable that the use of SVM with Binary Tree 4th grade polynomial kernel, the best in the Experiment 1, becomes the worst in the Experiment 2, where SVM Binary Tree RBF kernel is the best technique. We wanted to delve into this behaviour and, in Figure 1, it can be seen the relation between the number of categories and the accuracy of these techniques. We observe that as the categories increase, the performance of SVM techniques is getting worse, but not in the same way. It is therefore very important to select a suitable kernel if the number of categories is high.

It is noteworthy also to point out that the mistakes in the classification stage negatively affect the preparation of the summaries, since a news classified in a wrong way will be summarized in the second stage by means of an inadequate $valF$ function. That's why it's important to classify as best as possible. Even so, in both experiments the improvement that is obtained in the summary process is quite significant, so we can conclude that the applied methodology optimizes the process.

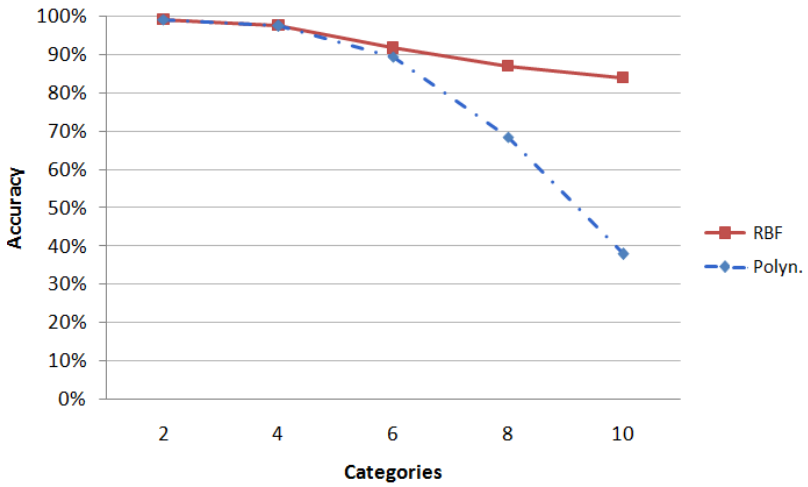

Fig. 1: Relation between number of categories and accuracy comparing SVM Binary Tree RBF versus SVM Binary Tree 4th degree Polynomial.

6 Conclusions and Future Work

In this work, we have presented a multilingual supervised learning methodology to generate automatic extractive summarizes. Our work focuses on the single-document general purpose extractive summaries, but with a significant difference: whereas other approaches considered a homogeneous corpus, we think that this aspect does not fit well to real scenarios, since within a set of documents, it is very usual to see different subsets with very different characteristics.

Our methodology can be applied in multiple working environments, with the advantage that the system, from a sample, is able to adapt to that context and specialize its way of making summaries. One of the key elements to make this work is the realization of an automatic categorization of source documents, to then, by solving an optimization problem, perform the adaptation of personalized summaries on each of the subsets resulting from such classifying them. The main contribution of this work is to improve the generation process of extractive summaries, in a general case, combining an automatic categorization of texts with performing a personalized adjustment of the summarization process according to this categorization. The utility of these kind of systems is clear for example for enhancing any generic documentation system, as we proposed in our initial works [30], or even for improving automatic infoboxes generation [31].

To evaluate our approach, we have applied this methodology in the field of news by developing a system specialized in summarizing news from media. We have performed the experiments over a real dataset, which is made available to other researchers on demand. The promising outcomes suggest that the methodology can be very useful in multiple scenarios and languages, an aspect that will be verified exhaustively in our following works. Also, our plans are to expand the features to consider texts, pointing to more linguistic and semantic issues, to enrich in that way the work done so far.

Acknowledgments

This research work has been supported by the CICYT project TIN2013-46238-C4-4-R, TIN2016-78011-C4-3-R (AEI/FEDER, UE), and DGA/FEDER. We want to thank Grupo Heraldo for their collaboration, and specially to Domingo Tardos and Susana Sangiao.

References

1. R. Brandow, K. Mitze, and L. F. Rau, "Automatic condensation of electronic publications by sentence selection," *Information Processing & Management*, vol. 31, no. 5, pp. 675–685, 1995.
2. Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, and Y. Yu, "Understanding and summarizing answers in community-based question answering services," in *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pp. 497–504, Association for Computational Linguistics, 2008.
3. N. Padhy, P. Mishra, and R. Panigrahi, "The survey of data mining applications and feature scope," *International Journal of Computer Science, Engineering and Information Technology*, vol. 2, no. 3, pp. 43–58, 2012.
4. V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 258–268, 2010.
5. C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*, pp. 71–78, Association for Computational Linguistics, 2003.
6. P. Lal and S. Ruger, "Extract-based summarization with simplification," in *Proceedings of the 2002 Workshop on Text Summarization (DUC'02)*, pp. 1–8, NIST, 2002.
7. W. Li, M. Wu, Q. Lu, W. Xu, and C. Yuan, "Extractive summarization using inter-and intra-event relevance," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING ACL'06)*, pp. 369–376, Association for Computational Linguistics, 2006.
8. A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining text data*, pp. 43–76, Springer, 2012.
9. H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.
10. H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.
11. J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pp. 68–73, ACM, 1995.
12. C.-Y. Lin, "Training a selection function for extraction," in *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, pp. 55–62, ACM, 1999.
13. J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, pp. 406–407, ACM, 2001.
14. M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization (AS'02)*, pp. 1–8, Association for Computational Linguistics, 2002.

15. K. M. Svore, L. Vanderwende, and C. J. Burges, "Enhancing single-document summarization by combining ranknet and third-party sources.," in *Proceedings of the 2007 Joing Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*, pp. 448–457, Association for Computational Linguistics, 2007.
16. R. Ferreira, F. Freitas, L. de Souza Cabral, R. D. Lins, R. Lima, G. Franca, S. J. Simske, and L. Favaro, "A context based text summarization system," in *Proceedings of the 11th IAPR International Workshop on Document Analysis Systems (DAS'14)*, pp. 66–70, IEEE Xplore, 2014.
17. Y. Chang, X. Wang, Q. Mei, and Y. Liu, "Towards twitter context summarization with user influence models," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM'13)*, pp. 527–536, ACM, 2013.
18. C.-L. Hwang and K. Yoon, *Multiple attribute decision making: methods and applications a state-of-the-art survey*, vol. 186. Springer Science & Business Media, 2012.
19. F. F. Bond, *An introduction to journalism, A survey of the fourth estate in all its forms*. Macmillan, 1954.
20. D. MacQuail, *Mass communication theory: an introduction*. Sage Publications, 1983.
21. K. Wolny-Zmorzyński and A. Kozieł, "Journalistic genology," *Media Studies*, vol. 54, pp. 1–16, 2013.
22. A. Bell, "The discourse structure of news stories," *Approaches to media discourse*, pp. 64–104, 1998.
23. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, pp. 137–142, Springer, 1998.
24. K.-S. Shin, T. S. Lee, and H. jung Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.
25. A. L. Garrido, O. Gomez, S. Ilarri, and E. Mena, "NASS: News Annotation Semantic System," in *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI'11)*, pp. 904–905, IEEE, 2011.
26. A. L. Garrido, O. Gomez, S. Ilarri, and E. Mena, "An experience developing a semantic annotation system in a media group," in *Proceedings of the 17th International Conference on Applications of Natural Language Processing to Information Systems (NLDB'12)*, pp. 333–338, Springer, 2012.
27. A. L. Garrido, M. G. Buey, S. Ilarri, and E. Mena, "GEO-NASS: A semantic tagging experience from geographical data on the media," in *Proceedings of the 17th East European Conference on Advances in Databases and Information Systems (ADBIS'13)*, pp. 56–69, Springer, 2013.
28. A. L. Garrido, M. G. Buey, S. Escudero, A. Peiro, S. Ilarri, and E. Mena, "The GENIE project-a semantic pipeline for automatic document categorisation.," in *Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST'14)*, pp. 161–171, SCITEPRESS, 2014.
29. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
30. A. L. Garrido, A. Peiro, and S. Ilarri, "Hypatia: An expert system proposal for documentation departments," in *Proceedings of the 12th International Symposium on Intelligent Systems and Informatics (SISY'14)*, pp. 315–320, IEEE, 2014.
31. A. L. Garrido, S. Ilarri, S. Sangiao, A. Gañan, A. Bean, and O. Cardiel, "NEREA: Named entity recognition and disambiguation exploiting local document repositories," in *Proceedings of the 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'16)*, pp. 1035–1042, IEEE, 2016.