A trainable algorithm for summarizing news stories

Joel Larocca Neto Alexandre D. Santos
Celso A.A. Kaestner Alex A. Freitas Julio C. Nievola
PUC-PR
PPGIA, CCET
Rua Imaculada Conceição, 1155
Curitiba, PR. 80215-901 - Brazil
{joel, kaestner, denes, alex, nievola}@ppgia.pucpr.br
http://www.ppgia.pucpr.br/~alex

Abstract

This work proposes a trainable system for summarizing news and obtaining an approximate argumentative structure of the source text. To achieve these goals we use several techniques and heuristics, such as detecting the main concepts in the text, connectivity between sentences, occurrence of proper nouns, anaphors, discourse markers and a binary-tree representation (due to the use of an agglomerative clustering algorithm). The proposed system was evaluated on a set of 800 documents.

1 Introduction

Text summarization is essentially the process of reducing the size of a text, yet preserving its information content. Producing a summary of any given text is a challenge that requires a full understanding of the text, which is beyond the state-of-the-art of computer science [Brandow 94], [Mitra 97]. However, there are several robust text-summarization systems that use statistical techniques and/or techniques based on superficial, domain-independent linguistics analyses.

The vast majority of current summarization systems perform an extractive summarization. This is a relatively simplified form of the summarization task where original sentences of the document are selected to be included in the summary according to predefined criteria. This approach reduces the need for a full understanding of the text and avoids the need for generating a new text (the summary) in natural language. Extractive summarization can be defined as the selection of a subset of sentences of the document that is representative of its content. This is typically done by ranking the original sentences and selecting the sentences with higher score. Although there is no guarantee that the summaries obtained by this approach will have narrative coherence, this approach is feasible by using current technology, and in any case the summaries can be useful.

Although many documents already contain a summary produced by the text's author, the automatic generation of summaries presents some unique advantages:

(a) It is possible to generate a summary with the size specified by the user, with the desired level of granularity - unlike manually-written summaries, which are static;

(b) It is possible to create links between a sentence of the summary and a corresponding block of sentences in the full text;

(c) It is possible to create user-focused summaries, selecting information relevant for any given user, rather than information considered relevant by the author.

This work proposes a machine learning-based trainable system for summarizing news. In addition, our system also outputs an approximate argumentative structure of the text.

This paper is organized as follows. Section 2 discusses related work on text summarization. Section 3 introduces our proposed system. Section 4 discusses computational results obtained by applying the proposed system on a set of 800 documents. Finally, section 5 concludes the paper.

2 Related Work

2.1 Machine Learning-based Text Summarization

The most influential project in the area of trainable summarizers was the seminal work of [Kupiec 95]. Kupiec proposed to cast summarization as a classification problem, where the goal is to discover a classification function that accepts as input a sentence and assigns it to one out of two classes: “summary” or “not-summary”. His system used five features, namely: an indication of whether or not the sentence length was below a prespecified (*sentence-length cutoff feature*); occurrence of cue words (*fixed phrase*), position of the sentence in the text and in the paragraph (*paragraph*), occurrence of frequent words (*thematic words*), and occurrence of words in capital letters, excluding the first word of a sentence and common abbreviations (*uppercase word*). He used Naive-Bayes as the trainable algorithm. This algorithm computed the probability of each sentence being in the summary, and the sentences with higher probability were selected. The data set contained 188 documents with extractive summaries produced by using a combination of two techniques: (a) automatic alignment of sentences with a manually-produced summary; (b) manual alignment performed by human judges. Kupiec did experiments with several subsets of features, and the best results were achieved with a subset containing three features: *paragraph*, *fixed-phrase* and *sentence length cutoff*. Using this feature subset, Naive-Bayes performing cross-validation achieved an accurate rate of 42% on the test set.

[Teufel 99] developed techniques for summarizing long texts, like magazine articles, with 20 or more pages. Teufel proposes a technique that selects for inclusion in the summary a subset of sentences that preserves some information about the general rhetoric structure of the text. In order to train the summarizer, seven features/heuristics were used. Four of them were similar to the features used by Kupiec. The three other features were: *indicator quality*, indicating the occurrence of meta-comments in the text; *indicator rhetorics*, modeling the rhetorical contribution of the phrases; and *header type*, specifying in which part of the text the sentence is included - e.g. “Introduction”, “Conclusion”, etc. The document base used was the CMP-LG, containing about 201 technical articles. In experiments with Naive-Bayes performing cross-validation, the best result - achieved using all features but *indicator rhetorics* - was 66% of accuracy on the test set.

[Mani 98] has used several machine learning techniques to discover features indicating the salience of a sentence. This work addressed the production of generic and user-focused summaries. Features were divided into three groups: locational, thematic and cohesion features. The document base used was the CMP-LG, also used by [Teufel 99], which contains summaries provided by the text’s author. The extractive summaries required for training were automatically generated as follows: the summary provided by the author is applied as a query to each sentence of the document. The similarity between the query and each sentence is computed, and the n sentences most similar to the query are selected for the summary, so producing fixed-length summaries (typically 10% or 20% of the total number of sentences). The experiments used three different algorithms, evaluated via 10-fold cross-validation. The best result for generic summaries was obtained by C4.5, which achieved an accuracy rate of 69%.

2.2 Summarization of News

[Strzalkowski 98] has proposed a system for summarizing news based on regularities in text organization, called DMS (Discourse Macro Structure). The basic idea is that a summary should reflect the components of the DMS. According to the author, news tend to be generated by using components that can be classified into two basic categories: a “what’s-the-news” category and an optional “background” category. The background category, if present, provides the context necessary for understanding the news. Relevant sentences are selected through several criteria, including the presence of words and phrases frequent in the text, title words, words occurring in the initial sentence of multiple paragraphs, paragraphs close to the beginning of the document, occurrence of proper names of people, places, organizations and certain “clue phrases” that indicate a main point of the text. Background sentences were selected through the presence of anaphors and external references, partial proper nouns, and presence of connectives of discourse - i.e. cohesive markers such as “Furthermore”, “But”, “Also”. In addition, background sentences tend to represent an isolated cluster with respect to terms used in the rest of the document, having low cohesion with the rest of the text [Salton 94].

2.3 Obtaining the argumentative structure of the text

The problem of obtaining the complete rhetoric structure of texts was addressed by [Marcu 99]. Marcu proposed a system based on the Rhetorical Structure Theory (RST), one of the most popular discourse theories in Natural Language Generation research. The central point of the theory is the notion of rhetoric relations, which are relations between two non-overlapping segments of text, called Nucleus and Satellite. A nucleus express the essential arguments of the text, unlike the satellites. A nucleus is also understandable by itself, regardless of its satellites, but vice-versa is not true. However, the automatic identification of such structures requires a deep understanding of the text and the application of this approach in a considerable volume of texts has yet to be shown.

[Ono 94] and [Kurohashi 94] proposed methods for an approximate identification of discourse structure by using only superficial text analysis techniques, such as detection of key expressions and occurrences of identical words or synonyms, and computation of the similarity between two sentences.

[Yaari 97] proposed a text segmentation method based on agglomerative clustering. The bottom-up Hierarchical Agglomerative Clustering algorithm is a widely used clustering method in information retrieval and linguistics.

This algorithm successively grows "coherent" segments by appending lexically related paragraphs.

The algorithm is implemented as follows:

```
1) Partition the text to elementary segments
2) While more than on segment left do
     Apply a proximity test to find the two most similar consecutive segments, $s_i$, $s_{i+1}$.
     Merge $s_i$, $s_{i+1}$ into one segment
   End While
```

The proximity test used is the cosine similarity between two text segments, using the vectorial representation. Unlike general agglomerative clustering applications, the proximity test is applied only to the two neighbors of a segment.

The output of the algorithm is a binary tree built through iterative merging of the most similar paragraphs.

[Mani 98b] compared the trees produced via agglomerative clustering with rhetorical structure [Marcu 99] on seven documents. The results show that the trees produced by agglomerative clustering have a coarse-grained segmentation (at the level of paragraph) of the topic structure of the text, whereas rhetorical structure trees have a fine-grained segmentation (at the level of clause). In general the trees generated by both approaches are very different, but occasionally there are some points of similarity.

3 The Proposed Approach

News represent an important field of application for automatic summarization. The goal of this work is the development of a trainable system for summarizing news and obtaining an approximate argumentative structure of the text, using some heuristics proposed by [Strzalkowski 98].

3.1 Obtaining an approximate structure of the text

Our proposed method produces an approximate structuring of the text combining ideas of several of the projects mentioned in section 2. More precisely, we combine the output of an agglomerative clustering algorithm with the detection of sentences that are either relevant (capturing the main ideas of the document) or background (containing additional, not essential information).

The detection of background sentences is based on the following criteria:

* The sentence contains few or no main concept(s) of the text;

* The sentence is located at a shallow position (i.e. close to the root) in the tree produced by agglomerative clustering, since it has low cohesion with the other sentences. We consider a sentence position as “shallow” if its depth is less than or equal to half the total depth of the tree;

* The sentence contains anaphors or external references;

* There are discourse markers in the beginning of the sentence.

To illustrate the main ideas of our proposed method, consider the following text and its corresponding tree produced by agglomerative clustering in figure 1.

P1 *(0.29)* Old-Fashioned and Reliable

P2 *(0.48)* Nantucket Corp., Clipper, $695

P3 *(0.41)* A solid and unspectacular performer, Clipper is a good choice for developers who want totally self-sufficient applications needing no run-time support.

P4 *(0.04)* Clipper operates in the two-step compile-and-link cycle familiar to programmers using traditional languages, such as C or COBOL.

P5 *(0.24)* In the application-development stage, Clipper is not as convenient as an interpreter, and its programs are not quite as fast as those of FoxBASE+.

P6 *(0.14)* HOWEVER, Clipper offers the advantage of stand-alone programs that need no run-time system.

P7 *(0.13)* ALTHOUGH Clipper pioneered many of the dBASE language enhancements now embraced by Ashton-Tate and other vendors, its implementation lags behind the others' in some ways.

P8 *(0.12)* FOR EXAMPLE, Clipper supports only one-dimensional arrays; dBASE IV and FoxBASE+ allow for two dimensions, and Quicksilver can handle as many as 255.

P9 *(0.41)* The product's two methods for creating menus are confusing and rather limited.

P10 *(0.16)* THE FIRST, which is similar to dBASE IV's syntax, requires the definition of each menu item in a separate program statement; the other method defines menu prompts as the elements of an array.

P11 *(0.20)* Both methods create one-dimensional lists of options, and although nested menus are possible, automatic drop-downs are not.

P12 *(0.11)* Clipper has been around for some time, and many aftermarket products, such as function libraries and help files, are available.

P13 *(0.21)* Any third-party enhancements for dBASE-compatible products are sure to include a Clipper-compatible version.

P14 *(0.23)* Nantucket Corp. of Los Angeles can be reached at (213) 390-7923.

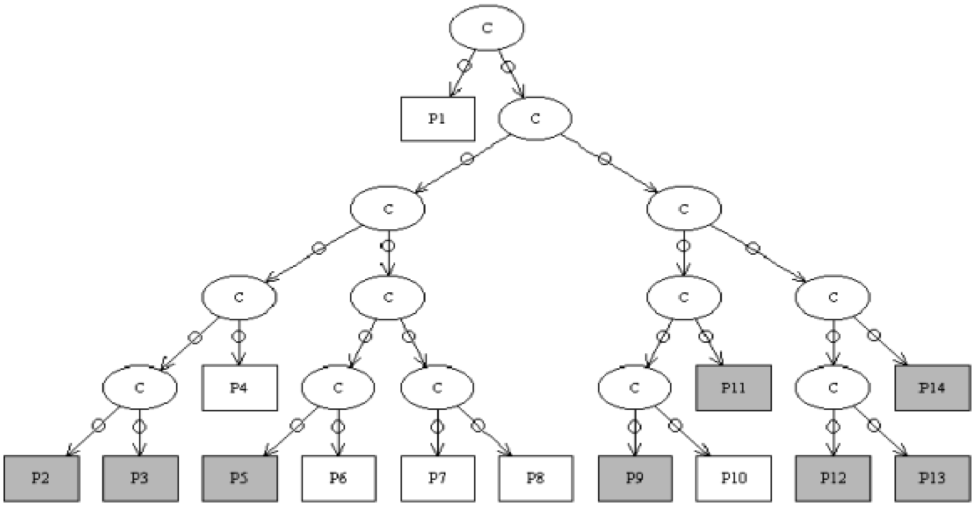

Figure 1 – Binary Tree produced by agglomerative clustering

The real-valued number between brackets is the degree to which the sentence captures the main concepts of the document.

In this example the following sentences would be considered background:

* 1 - low depth in the tree

* 4 - low relevance (0.04) with respect to the main concepts of the document

* 6, 7 and 8 - presence of discourse markers: “however”, “although” and “for example”

* 10 - presence of anaphors in the expression “the first”.

Therefore, only the sentences represented by a grey box in the figure would be considered essential. Note that the clusters represented in the figure were formed by grouping sentences according to the main ideas of the text, as follows:

* Group P2-P3 provides an overview of the product;

* Group P5-P8 compares the Clipper language with other languages such as FoxBASE+;

* Group P9-P11 discusses methods of menu creation;

* Group P12-P14 discusses the use of libraries and third party tools.

Although the results achieved by the proposed method are not nearly enough similar to rhetorical structure trees, the representation obtained by the method eliminated intuitively irrelevant sentences from the text and grouped together sentences containing similar ideas.

3.2 A trainable system for summarizing news

Using an approximate representation of text structure we can select a set of features to be used by an automatic summarization system. The features used in our system are as follows:

3.2.1 Main concepts indicator

This feature indicates whether or not the sentence captures the main concepts of the document. The main concepts of the document are in general represented as a set of words/phrases, which can be obtained in several different ways. The simplest method is to detect words with high frequency in the document - i.e. high tf (term frequency). Alternatively, one can detect words with high value of tf-idf (term frequency - inverse document frequency), which corresponds to detecting words that not only have high frequency in the current document but also are relatively rare in a large collection of documents. In [Lin 95] the central concepts of the text were obtained through the generalization relationships found in WordNet. [Turney 00] proposed the system GenEx, formed by two components: the genetic algorithm Genitor, which maximizes the performance in training data, and the algorithm Extractor, which obtains a list of key phrases from a document. In our system we propose a simpler idea: we restrict the analysis of the text only to individual words, rather than phrases. Since the vast majority of significant words in a document are nouns, we propose the following procedure to identify relevant words:

(a) All sentences are analyzed by a part-of-speech software. We have used Brill [Brill 92], a POS software widely used in the literature;

(b) All nouns in the document are considered as terms, removing duplicates - i.e. each noun corresponds to just one term, regardless of its number of occurrences in the text.

(c) For each term, we count the number of sentences where the term occurs at least once;

(d) The 15 terms with largest value of the counter computed in step (c) are selected.

Despite the simplicity of the above procedure, analysis of our experiments have shown that it effectively obtains words that are representative of the documents' contents. In addition, we have analyzed the use of the Extractor software to obtain the 15 words/key phrases more relevant of each document, as will be discussed in section 4.

3.2.2 Occurrence of proper nouns

Occurrence of proper nouns of people, places and organizations represent clues of positive relevance of a sentence for the summary, especially in news texts. In order to detect proper nouns we decided to capitalize on the use of the Brill Part-of-Speech tagger, considering as proper nouns all words that were so identified by that software. Analyzes of our results have shown that the majority of proper nouns of people and places were correctly detected by Brill.

3.2.3 Occurrence of anaphors

Ocurrence of anaphors usually indicates the presence of additional information, not essential for the contents of the summary. The system used for detecting anaphors is similar to the one proposed by [Strzalkowski 98]: certain nouns and expressions that occur in the beginning of sentences (in the first six words) are used for detecting anaphors. It should be noted that the system is designed only for detecting anaphors, and *not* for resolving them (replacing the anaphor by the term which it refers to).

3.3.4 Occurrence of discourse markers in the beginning of sentence

Some markers that frequently occur in the beginning of discourse - such as “Because”, “Furthermore” and “Additionally” - are considered indicators of the presence of additional information, not essential for the summary. Our system used a list of about 150 common markers.

3.2.5 Connectivity of sentences

Once there is evidence that sentences not essential for the summary present low cohesion, we have included in our system a feature that indicates the degree of connectivity between sentences. To compute this feature we have used a procedure similar to the computation of Text Relationship Maps [Mitra 97], as follows:

- Each sentence is represented in vectorial form and is then applied as a query to the other sentences of the document, to compute the similarity between the query and each of the other sentences;
- The total similarity value of each sentence is obtained through the sum of the individual similarity values between that sentence and each of the other sentences;
- The total similarity value of each sentence is normalized by dividing the value computed in the previous step by the largest total similarity value among all sentences. This produces, for each sentence, a number between 0 and 1 which is the degree of connectivity of that sentence.

- Finally, the real-valued degrees of connectivity computed in the previous step are discretized into three categorical values: low, medium and high.

3.2.6 Sentence Depth in the Tree

This feature indicates the depth of the sentence in the tree produced by the agglomerative clustering algorithm, normalized by the entire tree depth. Intuitively, sentences located at shallow levels (close to the root) of the tree are associated with a lower degree of cohesion with the rest of the text, and so they probably represent non-essential information.

3.2.7 Position in the Tree

Locational features are very important in summarization systems, especially in trainable ones. In [Mani 98a] all the organizational structure of the text was used - paragraphs, sections, subsections, etc. Although we may assume that this kind of information is available in semi-structured texts, their use reduces the generality of the summarization system. In other words, this kind of information is not available in commonplace, non-structured text. We want our summarization system to be as general-purpose (concerning the type of the document being summarized) as possible. Hence, instead of relying on structural information available only in semi-structured texts, we have decided to obtain locational information from the tree produced by agglomerative clustering. This approach has the advantage that this kind of tree can be produced for any text, even non-structured ones.

We use the path from the root of the tree to the selected sentence, only considering the first 4 nodes. The possible values for each one of 4 nodes are: left, right and none.

Note that the locational information that we get from the tree is richer than the relative position of the sentence in the document, since the tree structure reflects information of cohesion between sentences, which corresponds to an approximate “topical division” of the text into four levels.

Using the above-described features, each document sentence is represented in our system as follows:

Occurrence of proper nouns	Main concepts indicator	Occurrence of anaphors	Occurrence of discourse markers in the beginning of sentence	Connectivity of sentences	Sentence Depth in the Tree	First Level Position in the Tree	Second Level Position in the Tree	Third Level Position in the Tree	Fouth Level Position in the Tree	Sentence in Summary
TRUE	High	FALSE	TRUE	High	Low	Left	Right	Left	Left	YES
TRUE	Low	FALSE	FALSE	Low	High	Right	Left	Left	None	NO

Table 1 – Features used in our system

4 Results

The system was trained and tested using Ziff-Davis texts from the TIPSTER’s document base [Harman 94]. The base consists of texts of magazines about computers, hardware, software, etc. Among the available texts, about 33,658 contain a summary provided by the text’s author. Text sizes are considerably diverse: from 2 kbytes to more than 64 kbytes. For our experiments we have used only texts whose size is between 20 to 25 kbytes, which constitute a set of 900 documents. We have partitioned this set into two document bases: an *initial-base* with 100 documents and a *final-base* with 800 documents. The former was used for some preliminary experiments, including the selection of a good feature subset. Once the feature subset was selected and some system parameters were set, we have performed the final experiments - whose results are reported in this section - using the *final-base*.

The manual production of extractive summaries is a very expensive task, with respect to both resources and time. It is important to avoid this bottleneck in our experiments, since we use a large document base. To achieve this goal we have opted for the automatic generation of extractive summaries proposed by [Mani 98a], as described in section 2. As mentioned in that section, this technique produces fixed-size summaries - typically 10% or 20% of the total number of sentences. Although there is no guarantee that the summaries produced by this technique have the same contents as the summary produced by the text's author, we have empirically observed that the automatically-produced summaries are well correlated with their manual counterpart. One example of both kinds of summary produced for a given text is as follows:

AUTHOR SUMMARY

The Securities and Exchange Commission (SEC) is completing the testing of its Electronic Data Gathering, Analysis and Retrieval System (EDGAR).
The agency will spend $12 million over the next four years for office automation services for its 2,400 employees.
As a result, all of its employees currently have microcomputers which are connected to local area networks.
The networks are linked to 56K-bps X.25 packet-switching wide-area networks provided by British Telecom Tymnet.
Employees will be able to access mainframe computers at the regional SEC offices in New York, Boston, Denver, Los Angeles and Atlanta from their desktops.
LANs running the NetWare operating system from Novell Inc will interconnect SEC employees to nine regional centers, and by May 1992 all data will be transferred through the packet-switched FTS 2000 network.

EXTRACTIVE SUMMARY

The Electronic Data Gathering, Analysis and Retrieval System, now in the operational testing phase, has been a hard-won success for the SEC.
To keep pace, the SEC plans to spend about $12 million over four years to bring office automation services to its 2,400 employees nationwide, Fogash said.
During the past few years the commission has bought microcomputers for all its employees and begun setting up integrated local area networks.
Five regional offices -- in New York, Boston, Denver, Los Angeles and Atlanta -- have LANs connected to Washington headquarters by a 56-kilobit\sec wide area network.
SEC plans to have Novell Inc. NetWare LANs in all nine regions interconnected by the end of this year and will move its packet-switched communications to FTS 2000 by next May, Fogash said.

For some kinds of text, such as interviews or very short texts, this automatic generation of extractive summaries leaves a lot to be desired. However, this is a limitation of extractive summaries in general, regardless of how they are generated, since extractive summarization does not involve the production of new text (which could improve the summary quality).

In both the preliminary experiments with the *initial-base* and the final experiments with the *final-base*, all results were obtained using a 10-fold cross-validation procedure. We have used two classification algorithms: Naive-Bayes and C4.5 [Quinlan 93]. The performance of these algorithms was compared against the baseline strategy of selecting the first n sentences of the document, where n is the number of sentences selected for the summary (by both Naive-Bayes and C4.5). In previous work [Brandow 94], [Mitra 97] the summaries produced by such a simple baseline strategy were better than some more elaborated techniques. This simple baseline strategy tends to have good performance particularly in newspaper-style text, where the most important information is often presented, in a condensed form, in the first few sentences of the document.

In our experiments we have produced fixed-size summaries with 10% or 20% of the sentences of the document. In order to produce fixed-size summaries we have used the following strategy for each classifier:

* In the case of Naive-Bayes, which computes the probability that each sentence belongs to the summary, the n sentences with largest value of this probability are selected for the summary;

* In the case of C4.5 - which in its default form outputs only the predicted class, and not the class probabilities - we have used the -p option of this tool, which generates soft-threshold decision trees providing an estimate of the class probabilities [Turney 00]. Our system selects the n sentences with largest value of this probability to be included in the summary.

Another problem we had to deal with in our experiments was the problem of unbalanced classes. In our case, only 10% or 20% of the examples belonged to the positive class (corresponding to a sentence included in the summary). In highly-skewed class distribution problems, such as our experiments when the summaries contained only 10% of the document sentences, several rule induction algorithms - including C4.5 - tend to predict almost always the negative class (sentence not included in the summary), since this prediction can easily lead to a high accuracy rate (90% and 80% in the case of our experiments). We have experimented with two solutions for this problem, namely:

* Majority-class removal: All positive-class (minority-class) examples are kept in the training set, but most negative-class (majority-class) examples are removed from the training set, so that the class distribution in the training set becomes approximately 50%-50%. A similar technique is used by [Mani 98a]. This technique has the disadvantage of discarding potentially useful data.

* Minority-class replication: Instead of removing negative-class (majority-class) examples, this technique replicates positive-class (minority-class) examples, so that the class distribution in the training set again becomes approximately 50%-50%. For instance, if the original class distribution was 10%-90%, after replication there will be nine copies of each of the original minority-class examples. Although this technique has the disadvantage of producing a larger training set with redundant data, which increases processing time, it avoids the danger of discarding potentially useful data, and so it is expected to lead to better predictive accuracy.

As usual in information retrieval, predictive accuracy is evaluated in terms of recall and precision. In the case of our experiments (and other experiments on trainable summarizers producing fixed-length summaries), however, the number of sentences predicted to belong to the summary equals the number of sentences actually in the summary, by definition. In this case precision = recall = accuracy rate, so hereafter we simply refer to the accuracy rate of the summarizer on the test set, for short.

4.1 Preliminary Experiments (100 documents)

The results of our preliminary experiments are reported in the below table:

Baseline	Naive-Bayes	C4.5 with Minority-class replication	C4.5 with Majority-class removal
22.70%	**38.49%**	33.82%	26.60%

Table 2 - Initial-base results for 10% summaries using all features

Naive-Bayes achieved the highest accuracy rate, among the four summarization methods. Both Naive-Bayes and C4.5 achieved accuracy rates higher than the baseline strategy. Out of the two strategies used for coping with unbalanced classes, minority-class replication produced better results.

In order to select a subset of relevant features for classification, among all features discussed in section 3.2, we have run C4.5 seven times. In each of these runs C4.5 had access to only one of the seven features discussed in section 3.2. (In all runs C4.5 also had access to the class attribute, of course.) Only three out of those seven features had a predictive accuracy higher than the baseline summarizer, namely:

* occurrence of proper nouns - accuracy of 22.9%
* connectivity of sentences - accuracy of 36.9%
* main concepts indicator - accuracy of 29.2%

In addition, we ran another experiment where the third above feature was modified to consider 15 keywords extracted from the document by the Extractor software. The result achieved by C4.5 using this feature only was:

* main concepts indicator - accuracy of 24.1%

Surprisingly, although the Extractor software is a much more elaborated method than the simple keyword-extraction procedure proposed in section 3.2.1, the latter led to better results than the former.

We then ran an experiment using only the three highest-performance features (occurrence of proper nouns, connectivity of sentences, main concepts indicator) and selecting 10% of sentences for the summary. The results are reported in the below table.

Baseline	Naive-Bayes	C4.5 with Minority-class replication	C4.5 with Majority-class removal
22.70%	38.38%	**39.55%**	39.46%

Table 3 - Initial-base results for 10% summaries using the three highest-performance features

Comparing Table 2 with Table 3, one can see that the performance of Naive-Bayes with the three selected features was almost the same as with all features. However, the performance of C4.5 with the three selected features was considerably better than with all features. When only the three selected features are used, Naive-Bayes and C4.5 achieve similar predictive accuracy rates.

We also ran another experiment using only the three above-mentioned features, but this time selecting 20% of sentences for the summary. The results are reported in the below table.

Baseline	Naive-Bayes	C4.5 with Minority-class replication	C4.5 with Majority-class removal
31.70%	46.94%	49.19%	**49.32%**

Table 4 - Initial-base results for 20% summaries using the three highest-performance features

The decision tree produced by C4.5 for summaries with 20% of sentences is extremely simple, as can be seen below.

```
SentenceConnectivity = medium: true (13056.0/4901.5)
SentenceConnectivity = low: true (5482.0/707.5)
SentenceConnectivity = high:
|   MainConcepts = high: false (7962.0/1910.5)
|   MainConcepts = medium: false (942.0/320.5)
|   MainConcepts = low:
|   |   ProperNames = TRUE: true (57.0/22.5)
|   |   ProperNames = FALSE: false (64.0/30.5)
```

Apparently, high-cohesion sentences are evidence that the sentence should *not* be included in the summary. This is surprising, since there are many techniques that select for the summary sentences with a high degree of cohesion [Mitra 97], [Barzilay 97]. However, these results are not entirely conclusive, because the summaries of our training base were not produced by humans. In addition, it is possible that these results reflect a tendency of the document base used in the experiments.

4.2 Final Experiments (800 documents)

The preliminary experiments reported in the previous section were useful for select the best combination of features. Once these parameters were set we run our final set of experiments, whose results are reported in the below table.

Experiment	Baseline	Naive-Bayes	C4.5 with Minority-class replication	C4.5 with Majority-class removal
10% - all Features	22.50%	37.27%	**37.92%**	36.69%
10% - three highest-performance features	22.50%	36.32%	**37.99%**	37.66%
20% - three highest-performance features	31.70%	47.79%	**50.60%**	**50.60%**

Table 5 - Final-base results

5 Conclusion

This work has proposed a trainable system that automatically summarizes news and obtains an approximate argumentative structure of their text. The system was evaluated on a base of 800 documents, which is considerably larger than the document base used in most of the literature on trainable summarizers. When producing a summary with 20% of sentences of the source document, our system achieved an accuracy rate of 50.6% using C4.5 as the sentence classifier.

Surprisingly, the use of only three features led to a performance similar or superior to the use of all seven features described in section 3.2. In particular, the features obtained from the tree produced by agglomerative clustering, occurrence of anaphors and discourse markers in the beginning of sentences were considered of little relevance for the summarization task. A possible explanation for this is that the summaries of our training base were automatically produced, rather than being produced by a human. This fact probably reduces the importance of features that evaluate the readability/coherence of the summary, such as occurrence of anaphors and discourse markers. A human judge would hardly select a sentence with an anaphor to be included in the summary - unless the previous sentence was also selected. By contrast, automatically-produced summaries do not have this restriction, they can easily contain sentences with anaphors.

In our future work we intend to evaluate the performance of the system on documents with summaries produced by human judges.

6 References

[Barzilay 97] Barzilay, R. ; Elahad, M. Using Lexical Chains for Text Summarization. In Mani, I. e Maybury, M. T., eds., In *Proceedings of the ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*. Association of Computional Linguistics. 1997.

[Brandow 94] Brandow, R. ; Mitze, K. , Rau, L. Automatic condensation of eletronic publications by sentence selection. *Information Processing and Management*, 31, 5 , 675-685. 1994.

[Brill 92] Brill, E. A simple rule-based part-of-speech tagger. In *Proccedings of the Third Conference on Applied Computational Linguistics*.Association for Computational Linguistics. 1992.

[Edmundson 69] Edmundson, H. P. new methods in automatic extracting. *Journal of the Association for Computing Machinery* 16(2):2644-285. 1969.

[Harman 94] Harman, D. Data Preparation. In R. Merchant, editor, *The Proceedings of the TIPSTER Text Program Phase I*. Morgan Kaufmann Publishing Co. 1994.

[Kupiec 95] Kupiec, J. ; Pedersen, J. O.; Chen, F. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference, Association of Computing Machinery, Special Interest Group Information Retrieval*, 68-73. 1995.

[Kurohashi 94] Kurohashi, S. ; Nagao, M. Automatic Detection of Discourse Structure by Checking Surface Information in Sentences. *Proceedings of COLING 94 - Fifteenth International Conference on Computational Linguistics*, Vol.2, pp.1123-1127. 1994.

[Lin 95] Lin, C. Y. Knowledge-based automatic topic identification. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, pp 308-310. 1995.

[Luhn 58] Luhn, H. The automatic Creation of Literature Abstracts. *IBM Journal of Research and Development* 2(92):159-165. 1958.

[Mani 98a] Mani, I.; Bloedorn, E. Machine Learning of Generic and User-Focused Summarization. In *Proceedings of the Fifteenth National Conference on AI (AAAI-98)*, 821-826, 1998

[Mani 98b] Mani, I.; Bloedorn, E ; Gates, B. Using Cohesion and Coherence Models for Text Summarization. 1998 *AAAI Symposium Technical Report SS-989-06*. AAAI Press. 1998.

[Marcu 99] Marcu, D. Discourse trees are good indicators of importance in text. In I. Mani and M. Maybury editors, *Advances in Automatic Text Summarization*, pages 123-136, The MIT Press. 1999.

[Mitra 97] Mitra, M. ; Singhal, A. ; Buckley, C. Automatic Text Summarization by Paragraph Extraction. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain. 1997.

[Ono 94] Ono, K. ; Sumita, K. ; Miike, S. Abstract Generation Based On Rhetorical Structure Extraction. *Proceedings of COLING 94 - Fifteenth International Conference on Computational Linguistics* . 1994.

[Quinlan 92] Quinlan, J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, Sao Mateo, CA. 1992.

[Salton 94] Salton, G. ; Allan, J. ; Buckley, C. ; Singhal, A. Automatic analysis, theme generation, and summarization of machine readable texts. *Science* 264:1412-1426. 1994.

[Strzalkowski 99] Strzalkowski , T. ; Stein, G. ; Wang, J. ; Wise, B. A Robust Practical Text Summarizer. In I. Mani and M. Maybury editors, *Advances in Automatic Text Summarization.* The MIT Press. 1999.

[Teufel 99] Teufel, S. ; Moens, M. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In: I. Mani, M. Maybury (eds.), *Advances in automatic Text Summarization*, MIT Press, 1999

[Turney 00] Turney, P.D. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2 (4). 2000.

[Yaari 97] Yaari, Y. Segmentation of Expository Texts by Hierarchical Agglomerative Clustering. *Technical Report*, Bar-Ilan University, Israel. 1997.