2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 103-114

Sentence-based Text Summarization : Modelling and Evaluation

Iason Demiros[1,2] , Harris Papageorgiou[1] and Stelios Piperidis[1,2]

[1] Institute for Language and Speech Processing
Artemidos 6 and Epidavrou
Athens 15125, Greece
Tel: +301- 6875 300, fax: +301 - 6852 6202
[2]National Technical University of Athens
{iason, xaris, spip}@ilsp.gr

Abstract. This paper addresses the problem of creating a summary by extracting a set of sentences that are likely to represent the content of a document. A small scale experiment is conducted leading to the compilation of an evaluation corpus for the Greek language. Three models of sentence extraction are then described, along the lines of shallow linguistic analysis, feature combination and machine learning. The first model is based on term extraction and statistical filtering. With the second model we approximate the target classification function using a linear regression, where the regression coefficients are selected by applying an Information Gain criterion to the individual features over the training corpus. The third model is a lazy learning method that generalizes on a new instance over previously stored examples. Evaluation reported in the paper shows that all methods outperform the lead baseline and they could be used for rapid light information retrieval-oriented summarization.

1 Introduction

In an increasingly information-laden world where unstructured text data are the predominant type of data stored online, systems that can filter and condense data so that only relevant information reaches the decision maker, have become the focus of considerable interest and investment. A summary can be defined as a synopsis of the content of a document by distilling the most important information for a particular user and task. Early systems were characterized by a shallow approach such as exploiting term frequency [13] or cue and location features [6]. The 1980's enjoyed a renaissance in the field, with research based on AI techniques such as scripts [12], logic and production rules [8] and hybrid approaches. Recent work has focused on extracts rather than abstracts, a trend that could change as natural language generation and fact extraction technologies are becoming more robust. Corpus-based systems follow up classical approaches. They combine the calculation of corpus statistics in a learning framework. [11] has developed a Bayesian classifier, [18] combines individual features with the Dempster-Shafer rule while [2] combines features by the Bayesian rule. Discourse Model structures are exploited in [14] and [23], and lexical chains in [3]. The difficulties inherent in evaluating the quality of a summary are discussed in [15], [17] and [7].

The aim of the summarization systems that we have developed is to extract the most important sentences of the text by a set of metrics that we define. We believe that a reductive transformation of the source text to summary by sentence selection rather than a full understanding of the text by parsing to logical form or the exploration of its discourse structure, is an adequate framework to apply and evaluate our systems. We focus on environments where indicative, information retrieval oriented summaries are useful, bearing in mind that without an intermediate source processing and possibly a full text interpretation, part of the important content might be missed [10]. The first summarization machine that we present, called a Term-based Statistical Summarizer (TSS), incorporates shallow linguistic processing for term extraction and statistical filtering through a general corpus. Learning the linear regression coefficients on a feature set leads to the second idea presented in this paper: a Regression-based Learning Summarizer (RLS). Finally, within the Memory-based Learning framework we construct a Memory-based Learning Summarizer (MLS). The evaluation of a summarization system is a key part of any such effort. In the paper we describe an experiment that we have conducted in order to compile a manually annotated summarization corpus that is used both for training our machines and for evaluation.

2 Experimental Design

A corpus of summaries at various levels of compression was required for training and evaluating the summarization methods that we propose in Section 3. We conducted an experiment at two compression levels and compiled a small corpus of 10 documents annotated by 26 subjects. The documents covered a variety of financial and political topics.

2.1 Dataset Properties and Annotation Procedures

A total of 26 graduate students and researchers from various disciplines participated in the study. They represented two teams of 13 subjects selected according to uniform criteria such as their age, background and gender. Each annotator processed 5 documents that were randomly distributed to two teams. As a result, our evaluation dataset consisted of 10 documents, each one abstracted 13 times. We also collected an optional qualitative simulation of the abstracting mechanism, from any annotator that was able to deliver such a description.

The documents were selected from the Greek financial press covering a variety of topics (international and national news, political articles, business news and commentary) and their size varied from 10 to 40 sentences giving a total size of 232 sentences for the whole corpus. Since we conducted a relatively small-scale experiment, we aimed at a uniform representativity of the compiled corpus in terms of the basic parameters such as the domain, the topic ambiguity, the targeted audience and the document length. Only the textual portion of each text was presented to the user. Any specific format was eliminated, the title was indexed as the first sentence in each document but the annotators were asked not to include it in their summaries. Each annotator was asked to carry out two experiments: to produce a summary at 10% and at 20% of the length of the full text, in number of sentences. The aforementioned compression levels are somehow standard for the "ideal" summary baseline evaluation of automatic summarization systems ([15], [7]).

2.2 Analysis and Results

2.2.1 Annotator Agreement

We used the positive agreement between two subjects as a metric for the overlap of their extracts. Since the number of annotators was rather high, percent agreement indicating the subject agreement with the majority opinion, including both the decision to extract and not to extract a sentence ([14]), would yield a high yet not indicative score in the experiment. Agreement between subjects is a metric reported to fall in the range between 25% to 90%. The main reasons for this significant deviation are to be interpreted by the different settings of each experiment, including the definition of the agreement metric. Table 1 presents the average agreement between all annotators taken by two, the maximum agreement which is equal to the total number of sentences that should be selected, and the subject agreement score at the compression levels of 10% and 20%.

Table 1. Subject agreement in the experiment

Compression level	Average Agreement	Number of Sentences	Score
10%	9.81	26	38%
20%	25.35	52	49%

2.2.2 Research Hypothesis

Statistical significance of the null hypothesis that "the number of annotators selecting the same sentence in a document is random", has been largely studied in [17] and [24]. Using standard methods to determine if the null assertion is reasonable and to what degree of certainty, they all reached the same conclusion: agreement between human abstractors is much higher than would be expected by chance. The difference between the hypothesis test procedures arise from the assumptions that the researchers are willing to make for the data sample: Z-test, t-test, F-test, Levene-test, Cochran-test, these can all be applied to evaluate the hypothesis. By applying the t-test to the 10 documents of the corpus we have shown that the probability of the statistic with df = 9 degrees of freedom is very low, leading to the conclusion that the agreement among annotators is highly significant. Table 2 summarizes the results of our test.

Table 2. Chi-square corpus statistics

Compression level	Mean agreement	Standard Deviation	χ^2 probability
10%	0.39	0.0987	0.00001
20%	0.48	0.0729	0.002

2.2.3 Virtual Summaries

Sentences in each document of the corpus were ranked by the absolute number of times they were selected from the annotators. As a result, all automatic ranking summarizers could be evaluated at any level of compression, be it 10%, 20% or any other. For the purpose of evaluating our abstraction machines we have retained the following summaries for each document:

- the 10% and the 20% majority summaries, created by selecting the sentences in descending order of ranking at the corresponding compression level.
- the 10% and the 20% relevance summaries, created by selecting all the sentences in descending order of ranking at the corresponding compression level, that were selected at least once by some annotator.

Table 3 details an example document from the training corpus.

Table 3. Example virtual summaries

Doc name	vimad675 (26 sentences)
10% majority	*3, 15, 22* (3/26)
20% majority	*15, 20, 3, 14, 21, 22* (6/26)
10% relevance	*3, 15, 22, 7, 14, 20, 8, 21, 5, 23, 11, 17* (12/26)
20% relevance	*15, 20, 3, 14, 21, 22, 8, 9, 17, 7, 10, 5, 11,12, 6, 23* (16/26)

Numbers in italics are sentence index numbers in the document, ranked by majority votes. For instance, sentences 3, 15 and 22 are selected for the 10% majority summary, among 12 sentences that had at least one vote and were all selected for the 10% relevance summary. Standard Precision, Recall and F measures were used to evaluate the automatic summarization systems that we propose in the next sections.

2.2.4 Smoothing the Extract

The problems of coherence and repetitiveness were exploited as a cognitive aspect of the experiment. We have used the algorithm presented in [19] to eliminate sentences that were extracted and yield a high partial matching score, but no such necessity was observed for the documents of the training corpus. That is, no two among the selected sentences bore resemblance to a high degree. Applying linguistic processing to resolve co-referential relations, although being indispensable in a discourse analysis, was considered an extremely costly solution. Although anaphoric expressions existed in the extracts, they did not harm the intelligibility except in one case out of 10 documents.

3 Automatic Text Summarization Machines

3.1 Term-based Statistical Summarizer (TSS)

Despite termhood vagueness, the identification and extraction of terms is one of the best understood and most robust Natural Language Processing (NLP) technologies. From the point of view of operational Information Retrieval (IR) and Information Extraction (IE) systems, the mapping from the document to a term set is a commonplace representation for content and domain characterization. Recent research has shown that relatively simple natural language analysis methods such as some form of partial parsing to match the text against a set of patterns, require less complex structures and sophisticated knowledge bases, which are not really important when considering the average information seeker but are essential in the area of domain-dependent large scale systems. Our goal will be to find those terms that are characteristic of a particular document.

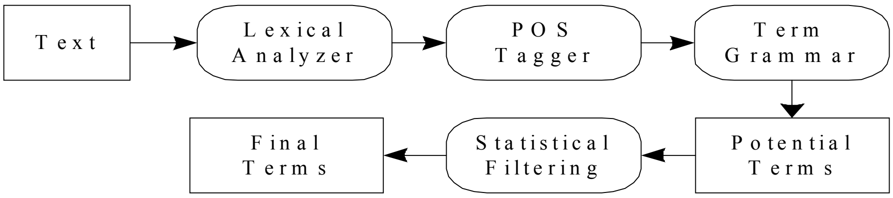
3.1.1 Term Extraction

The term extraction process is a hybrid one, and operates in four pipelined stages:

- Lexical analysis and sentence boundary identification
- morphosyntactic annotation
- shallow parsing using a regular expression-based, term pattern grammar
- statistical filtering in order to remove grammar-extracted terms lacking statistical evidence

Figure 1 illustrates the processing stages of term extraction:

Figure 1. Term Extraction overall flow



3.1.2 Pattern Grammar

The pattern grammar used in the syntactic analysis is a subset of pattern rules presented in [9], that cover a great part of the Greek terminology. The term grammar consists of a set of rules recognising one to three-word terms. Each rule was converted to a non-deterministic finite state automaton (NDFA).

3.1.3 Statistical Filtering

After the term grammar module has been applied, the extracted terms are statistically evaluated in order to remove items without adequate statistical evidence and thus reduce the overgeneration effect caused by pattern grammars. Statistical evaluation is performed using tf*idf , so that the frequency of the term in the domain is also taken into consideration. The tf*idf metric is a standard weight computation method which combines term frequency (TFi), the number of documents in a corpus[1] (N) and the number of documents (ni) that the term appears in:

$$tf * idf_i = TF_i \cdot \log \frac{N}{n_i} \quad (1)$$

3.1.4 Sentence Selection

We determine the weight of each sentence by computing the sum over all tf*idf values of the terms that were extracted in the previous step. The process provides a bias towards longer sentences which appears to be appropriate as analyzed in [24]. The final step of the method is sorting the sentences according to their weight and extracting the number of sentences corresponding to the 10% abstract and the 20% abstract respectively in input text order.

3.1.5 TSS Evaluation

The first abstracting machine that we have developed is an extension of the system described in [24], in order to calculate sentence weights based on terms rather than on content words. Table 4 shows the evaluation metrics for TSS. The prescribed nature of the system extracting the exact number of sentences corresponding to the given compression levels of 10% and 20% yields equal figures for Precision and Recall. We evaluate the system performance against the virtual summaries described in 2.2.3. Table 4 also shows the lead method evaluation against the virtual summaries. Numbers in bold are in percentage. Numbers in parentheses indicate the number of correctly extracted sentences against the maximum number that would lead to a 100% success. Both [24] and [11] consider the lead[2] method as their benchmark. TSS improves the lead method at all levels. The improvement is 15%(78%) against the 10%(20%) majority summary and 67%(55%) against the 10%(20%) relevance summary.

[1] We used the ILSP Greek financial and political press corpus of 1200 documents (500K words)

[2] Select the first-N sentences from the beginning of the document

Table 4. All methods against the baseline

	10% majority	20% majority	10% relevance	20% relevance
LEAD	**27**(7/26)	**27**(14/52)	**46**(12/26)	**58**(30/52)
	27(7/26)	**27**(14/52)	**46**(12/26)	**58**(30/52)
TSS	**31**(8/26)	**48**(25/52)	**77**(20/26)	**90**(47/52)
	31(8/26)	**48**(25/52)	**77**(20/26)	**90**(47/52)
RLS	**50**(13/26)	**56**(29/52)	**81**(21/26)	**92**(48/52)
	50(13/26)	**56**(29/52)	**81**(21/26)	**92**(48/52)
MLS	**15**(15/98)	**29**(28/98)	**57**(56/98)	**78**(76/98)
	58(15/26)	**54**(28/52)	**55**(56/101)	**51**(76/14)

3.2 Regression-based Learning Machine (RLS)

3.2.1 Learning Framework

The idea of developing a summarization system by combining the evidence from individual features has been extensively studied in the bibliography. Edmundson ([6]) introduces a scoring formula that encapsulates more than one diagnostic unit and has the form of a weighted sum:

$$Score = \sum_{i=1}^{n} w_i x_i \quad (2)$$

where w_i is the weight of the i-th feature, x_i is the specific score of the sentence for the i-th feature, n being the number of features. Automatic training of the weights w_i in a multiple linear regression analysis will be the core component of the RLS summarizer.

3.2.2 Feature Set

We settled on a version of the feature set that was introduced in [6].

tf*idf feature: The term extraction module that was described in TSS calculates the tf*idf score of each sentence as an integer number[3].

cue phrase feature: The cue dictionary in use was compiled by human annotators and refined according to linguistic criteria[4].

lead feature: We have implemented a fraction of the general location method that calls for parsing the skeleton of the document. After several experiments we have retained the lead-4 characteristic.

title feature: Based on the hypothesis that the title circumscribes the subject matter of the text, we match each sentence's content words to the title content words at the lemma level.

3.2.3 Weighting by Information Gain

The general computational problem that needs to be solved in multiple regression analysis is to fit a straight line to the instances in the training set. Each sentence is described in terms of the four feature values described in 3.2.2. The purpose is to

[3] The sentence length bias is not eliminated

[4] Our dictionary contains 50 indicator single and multi-word units

learn the relationship between the predictor features and the binary target function that selects a sentence to be part of the summary. The training set consists of 232 sentences labeled positively if they belong to the relevance summary (selected at least once by an annotator) and negatively, if not.

We use Information Gain ([20]) as a measure of the effectiveness of a feature in classifying the training data. The Information Gain G(S, F) of a feature F relative to a training set S, is defined as

$$G(S,F) = H(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} H(S_v) \quad (3)$$

where Values(F) is the set of all possible values for the feature F, S_v is the subset of S for which feature F has value v: $S_v = \{s \in S|F(s) = v\}$ and $H(S) = Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$ where p_i is the proportion of S belonging to class i. Table 5 shows the Information Gain values for the feature set of RLS calculated in the collection of the sentence examples. A normalized version of Information Gain (Gain Ratio) that eliminates the bias towards features with larger values has been applied. The weight of each feature in (2) is equal to the feature's Information Gain in the training corpus. Thus, we have estimated the regression coefficients representing the independent contribution of each feature to the prediction of the dependent variable, equivalently, to the decision whether a sentence will be selected for inclusion in the abstract. Each feature value is expressed by a normalization to its maximum value within the document. This way a common range of values is defined for the full feature set and the bias towards large values is smoothed[5].

Table 5. Feature values using Information Gain

Feature	**Tf*idf**	**Cue**	**lead**	**Title**
Information Gain	0.56	0.38	0.25	0.37

3.2.4 RLS Evaluation

RLS is the second machine we have developed, based on shallow linguistic processing and machine learning techniques. We determined the score of each sentence by using (2) and the weights figured in Table 5. The sentence selection procedure is identical to the one described in 3.1.5 for the TSS. The evaluation results are illustrated in Table 4.

Since the dataset was small, we used a cross validation strategy for evaluation by selecting documents for testing one at a time and using all other documents for training. Evaluation figures show a noticeable improvement in comparison to both TSS and the lead benchmark: at the 10%(20%) majority level the improvement is

[5] This way all values are scaled-down in the range [0-1], assigning 1 to the larger value of the particular feature in the document

61%(17%) against TSS and 85%(107%) against lead. The numbers at the relevance level are 5%(2%) and 76%(59%) respectively[6].

3.2.5 Remarks and Limitations

Multiple linear regression is a powerful concept learning method, but with several limitations. The linear relationship between variables can practically never be confirmed. It is assumed that the residuals (predicted minus observed values) are normally distributed, assumption that does not always hold true[7]. Finally, the major conceptual limitation of applying the regression technique in text summarization is that one can only ascertain relationships, but never be sure about the underlying causal mechanism.

It is important to provide a further justification for using the notion of Information Gain to calculate the regression coefficients. The concept of Maximum Entropy is widely used for the construction of stochastic models of natural language. From a Machine Learning point of view, a model with Maximum Entropy maximizes the likelihood of the training sample. Thus, it minimizes the squared error between the output hypothesis predictions and the training data ([16]). By consequence, our method can be considered as an indirect least squares estimator for the dependent variable that represents the decision to include a sentence in the abstract.

3.3 Memory-based Learning Machine (MLS)

3.3.1 Similarity-based Induction for Summarization

The main source of inspiration is the case-based approach to language ([1], [21]) in a phenomenological learning framework (true of the data but does not explain much of the machinery). We will test whether pattern association by direct reference to memory can be applied to the problem of sentence extraction.

Both TSS (3.1) and RLS (3.2) commit to a single hypothesis governing the entire paradigm space. Their approximation of the target function takes the form of reasoning by deduction. An alternative approach could be to use the set of stored experiences as the basis of answering questions about newly encountered instances. The potential uses of this form of reasoning, called Memory-based Reasoning, in language processing, are rapidly expanding.

3.3.2 Algorithm and Features

When classification of a test vector x_S is required, Memory-based algorithms retrieve and analyze the training data in a “local neighborhood” of x_S ([22]). The algo-

[6] We have experimented with relaxing the lead baseline by selecting each time a number of sentences equal to the compression levels of 10% and 20%. The figures to beat were very low and a uniform first-4 baseline was thought to be more indicative

[7] It is always a good idea, before drawing final conclusions, to review the distributions of the features of interest

rithm that has been used is a variation of the k-Nearest Neighbor algorithm. It learns the target function by assigning the test vector x_S to the hypothesis that is most frequently represented in the *k* nearest training examples to x_S. A formal analysis of the nearest neighbor presented in ([4]) is based on the following assumptions: the classified examples are independently and identically distributed (iid) and the sample size *N* is infinitely large. In order to approximate the discrete-valued function $f : \Re^n -> V = \{0,1\}$ each example $(x, f(x))$ is stored into the memory. Given a new instance to be classified, the value $f(x_n) = \arg\max_{v \in V} \sum_{i=1}^{k} \Delta(v, f(x_i))$ is calculated, where $x_1 ... x_k$ are the *k* training instances nearest to x_n and $\Delta(a,b)$ is the distance between the instances a, b.

For our experiments we have used TiMBL ([5]). The distance between two instances is the sum of the differences between their features, weighted by the Information Gain of each feature. The Modified Value Difference Metric described in ([22]) was used to determine the similarity between two values of a feature. The feature set is identical to the one described in 3.2.2 and the number of training instances nearest to x_n was fixed to 1.

3.3.3 MLS Evaluation and Discussion

A cross validation strategy for evaluation by selecting documents for testing one at a time and using all other documents for training was followed. The results of the Memory-based approach for summarization are shown in Table 4. During classification each sentence is presented to the system as a feature vector with four dimensions (tf*idf, cue, lead, title) and is subsequently selected or rejected for inclusion in the abstract according to the k-NN algorithm[8]. Since the binary-valued classification function does not have ranking capabilities, an average value of 42% of the input text is extracted (98 sentences out of 232). In order to cater for the relative strength of each sentence, all features are normalized by a division with their maximum values in the document. Naturally the normalization holds for the classified instances too. We have thus been able to capture the contextual dependency of each sentence on all the other sentences of the document and to calculate the distance between the relative values of two instances in their proper documents, instead of their absolute values.

Although k-NN is an effective inductive reasoning method, it is sensitive to a sparse dataset such as the one we have compiled and suffers from the curse of dimensionality when irrelevant features are present. Also, since no post-editing was applied to the output of the linguistic components such as the sentence recognizer, the morphosyntactic analyzer and the term extractor, we can expect an impact from the noisy examples in our corpus[9].

[8] The dataset consists of 232 sentences manually classified as described in previous sections

[9] The success of our tools is 95% for sentence recognition, 93% for POS tagging, 80% for lemmatization and 60% for term recognition

Table 6. F-values for all methods

F score	LEAD Maj	TSS maj	RLS maj	MLS maj	LEAD rel	TSS rel	RLS rel	MLS rel
10%	27	31(+15)	50(+85)	24(-11)	46	77(+67)	81(+76)	56(+22)
20%	27	48(+78)	56(+107)	37(+37)	58	90(+55)	92(+59)	62(+7)

4 Systems Comparison and Conclusion

A comparison of all methods that we have presented in this paper is given in Table 6. Numbers are in percent. Numbers in parentheses show each method improvement against the LEAD benchmark. The machines we have developed are grounded in a robust machine learning framework. The feature computation procedure requires shallow linguistic processing and statistical filtering. We have demonstrated the quality of the systems through experiments that involved a large number of human annotators. Significantly higher scores than the lead benchmark are obtained for all systems, at the 20% level. TSS performance at the 10% level is almost equivalent to the baseline, but MLS score at the same level is below the baseline, due to the limitations explained in the previous section. The regression machine performance is impressive, outperforming all other machines as well as the benchmark and being even above the average subject agreement figures. Many ideas concerning machine learning in text abstraction will be investigated in the future within a context-based strategy ([10]): correlation between the features and non-linear regression analysis, introduction of a new set of sophisticated psycholinguistic features and local probabilistic modeling of the target function, among others.

References

1. David Aha, Dennis Kibler, Marc Albert. 1991. Instance-Based Learning Algorithms. In *Machine Learning* 6, 37-66.
2. Chinatsu Aone, Mary Ellen Okurowski, James Gorlinsky. 1998. Trainable, Scalable Summarization Using Robust NLP and Machine Learning. In *Proceedings of COLING-ACL 1998*: 62-66.
3. Regina Barzilay and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of ACL'97 Workshop on Intelligent, Scalable Text Summarization*, 10-17.
4. T. Cover and P. Hart. 1967. Nearest Neighbor Pattern Classification. In *IEEE Transactions on Information Theory*, 13, 21-27.
5. Walter Daelemans, Jakub Javrel, Ko van der Sloot, Antal van den Bosch. 2000. TiMBL: Tilburg Memory Based Learner, version 3.0, reference manual. Technical Report ILK-9901, ILK, Tilburg University, 2000.
6. H. P. Edmundson. 1969. New methods in Automatic Extracting. In *Journal of the Association for Computing Machinery* 16(2):264-285
7. Thérèse Firmin and Michael J. Chrzanowski. 1998. Automatic Summarizing: Factors and Directions. In *Advances in Automatic Text Summarization*, Ed. I. Mani and M. Maybury, Cambridge MA: MIT Press, 1998, 325-336.
8. D. Fum, G. Guida, C. Tasso. 1985. Evaluating Importance: a step towards Text Summarization. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'85)*, 840-844.

9. Maria Gavrilidou, Penny Lambropoulou (1994). Report on the Constituent Grammar, *RENOS project, LREI- 62-048*, Athens
10. Karen Spärck Jones. 1998. Automatic Summarizing: Factors and Directions. In *Advances in Automatic Text Summarization*, Ed. I. Mani and M. Maybury, Cambridge MA: MIT Press, 1998, 1-12.
11. Julian Kupiec, Jan Pedersen, Francine Chen. 1995. A Trainable Document Summarizer. In Proceedings of the 18th ACM-SIGIR Conference, 88-97.
12. W. Lehnert, J. Black, B. Reiser. 1981. Summarizing narratives. In *7th International Joint Conference on Artificial Intelligence*. Vancouver, British Columbia.
13. H. Luhn. 1958. The Automatic Creation of Literature Abstracts. In *IBM Journal of Research and Development* 2(2):159-165.
14. Daniel Marcu. 1997. From Discourse Structures to Text Summaries. In *ACL/EACL-97 summarization workshop*, 82-88.
15. Vibhu O. Mittal, Mark Kantrowitz, Jade Goldstein, and Jaime Carbonell. 1999. Selecting Text Spans for Document Summaries: Heuristics and Metrics. In *Proceedings of AAAI-99*, pages 467-473, Orlando, FL, July 1999. AAAI.
16. Tom Mitchell. 1997. Machine Learning. McGraw-Hill.
17. Andrew Morris, George Kasper, Dennis Adams. 1992. The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance. In *Advances in Automatic Text Summarization*, Ed. I. Mani and M. Maybury, Cambridge MA: MIT Press, 1998, 305-323.
18. Sung Hyon Myaeng and Dong-Hyun Jang. 1997. Development and Evaluation of a Statistically-based Document Summarization System. . In *Advances in Automatic Text Summarization*, Ed. I. Mani and M. Maybury, Cambridge MA: MIT Press, 1998, 61-70.
19. Stelios Piperidis, Christos Malavazos, Ioannis Triantafyllou. 1999. A Multi-level Framework for Memory-Based Translation Aid Tools. In ASLIB: Translating and the Computer, 21st Conference, November 1999.
20. J. Ross Quinlan. 1993. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.
21. Royal Skousen. 1989. Analogical Modeling of Language. Dordrecht: Kluwer.
22. Craig Stanfill and David Waltz. 1986. Toward Memory-based reasoning. In *Communications of the ACM*, 29: 1212-1228.
23. Simone Teufel and Marc Moens. 1998. Argumentative Classification of Extracted Sentences as a First Step Towards Flexible Abstracting. In *Advances in Automatic Text Summarization*, Ed. I. Mani and M. Maybury, Cambridge MA: MIT Press, 1998, 155-171.
24. Klaus Zechner. 1996. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark, pp. 986-989.