Automatic Text Summarization Using Unsupervised and Semi-supervised Learning

Massih-Reza Amini and Patrick Gallinari

LIP6, University of Paris 6, Case 169, 4 Place Jussieu
F – 75252 Paris cedex 05, France
{amini,gallinari}@poleia.lip6.fr

Abstract. This paper investigates a new approach for unsupervised and semi-supervised learning. We show that this method is an instance of the Classification EM algorithm in the case of gaussian densities. Its originality is that it relies on a discriminant approach whereas classical methods for unsupervised and semi-supervised learning rely on density estimation. This idea is used to improve a generic document summarization system, it is evaluated on the Reuters news-wire corpus and compared to other strategies.

1 Introduction

Many machine learning approaches for information access require a large amount of supervision in the form of labeled training data. This paper discusses the use of unlabeled examples for the problem of text summarization.

Automated summarization dates back to the fifties [12]. The different attempts in this field have shown that human-quality text summarization was very complex since it encompasses discourse understanding, abstraction, and language generation [25]. Simpler approaches were explored which consist in extracting representative text-spans, using statistical techniques and/or techniques based on superficial domain-independent linguistic analyses. For these approaches, summarization can be defined as the selection of a subset of the document sentences which is representative of its content. This is typically done by ranking the document sentences and selecting those with higher score and with a minimum overlap. Most of the recent work in summarization uses this paradigm. Usually, sentences are used as text-span units but paragraphs have also been considered [18, 26]. The latter may sometimes appear more appealing since they contain more contextual information. Extraction based text summarization techniques can operate in two modes: generic summarization, which consists in abstracting the main ideas of a whole document and query-based summarization, which aims at abstracting the information relevant for a given query.

Our work takes the text-span extraction paradigm. It explores the use of unsupervised and semi-supervised learning techniques for improving automatic summarization methods. The proposed model could be used both for generic and query-based summaries. However for evaluation purposes we present results on a generic summarization task. Previous work on the application of machine learning techniques for summa-

L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp. 16-28, 2001.

rization [6, 8, 11, 13, 29] rely on the supervised learning paradigm. Such approaches usually need a training set of documents and associated summaries, which is used to label the document sentences as relevant or non-relevant for the summary. After training, these systems operate on unlabeled text by ranking the sentences of a new document according to their relevance for the summarization task.

The method that we use, to make the training of machine learning systems easier for this task, can be interpreted as an instance of the Classification EM algorithm (CEM) [5, 15] under the hypothesis of gaussian conditional class densities. However instead of estimating conditional densities, it is based on a discriminative approach for estimating directly posterior class probabilities and as such it can be used in a non parametric context. We present one algorithm upon on linear regression in order to compute posterior class probabilities.

The paper is organized as follows, we first make a brief review of semi-supervised techniques and recent work in text summarization (Sect. 2). We present the formal framework of our model and its interpretation as a CEM instance (Sect. 3). We then describe our approach to text summarization based on sentence segment extraction (Sect. 4). Finally we present a series of experiments (Sect. 5).

2 Related Work

Several innovative methods for automated document summarization have been explored over the last years, they exploit either statistical approaches [4, 26, 31] or linguistic approaches [9, 14, 22], and combinations of the two [2, 8]. We will focus here on a statistical approach to the problem and more precisely on the use of machine learning techniques.

From a machine learning perspective, summarization is typically a task for which there is a lot of unlabelled data and very few labeled texts so that semi-supervised learning seems well suited for the task. Early work for semi supervised learning dates back to the 70s. A review of the work done prior to 88 in the context of discriminant analysis may be found in [15]. Most approaches propose to adapt the EM algorithm for handling both labeled and unlabeled and to perform maximum likelihood estimation. Theoretical work mostly focuses on gaussian mixtures, but practical algorithms may be used for more general settings, as soon as the different statistics needed for EM may be estimated. More recently this idea has motivated the interest of the machine learning community and many papers now deal with this subject. For example [19] propose an algorithm which is a particular case of the general semi-supervised EM described in [15], and present an empirical evaluation for text classification. [16] adapt EM to the mixture of experts, [23] propose a Kernel Discriminant Analysis which can be used for semi-supervised classification.

The co-training paradigm [3] is also related to semi supervised training. Our approach bears similarities with the well-established decision directed technique, which has been used for many different applications in the field of adaptive signal processing [7].

For the text summarization task, some authors have proposed to use machine learning techniques. [11] and [29] consider the problem of sentence extraction as a classifi-

cation task. [11] propose a generic summarization model, which is based on a Naïve-Bayes classifier: each sentence is classified as relevant or non-relevant for the summary and those with highest score are selected. His system uses five features: an indication of whether or not the sentence length is below a specified threshold, occurrence of cue words, position of the sentence in the text and in the paragraph, occurrence of frequent words, and occurrence of words in capital letters, excluding common abbreviations.

[13] has used several machine learning techniques in order to discover features indicating the salience of a sentence. He addressed the production of generic and user-focused summaries. Features were divided into three groups: locational, thematic and cohesion features. The document database was CMP-LG also used in [29], which contains human summaries provided by the text author. The extractive summaries required for training were automatically generated as follows: the relevance of each document sentence with respect to the human summary is computed, highest score sentences are retained, for building the extractive summary. This model can be considered both as a generic and a query-based text summarizer.

[6] present an algorithm which generates a summary by extracting sentence segments in order to increase the summary concision. Each segment is represented by a set of predefined features such as its location, the average term frequencies of words occurring in the segment, the number of title words in the segment. Then they compare three supervised learning algorithms: C4.5, Naïve-Bayes and neural networks. Their conclusion is that all three methods successfully completed the task by generating reasonable summaries.

3 Model

In this section, we introduce an algorithm for performing unsupervised and semi-supervised learning. This is an iterative method that is reminiscent of the EM algorithm. At each iteration, it makes use of a regression model for estimating posterior probabilities that are then used for assigning patterns to classes or clusters. The unsupervised version of the algorithm may be used for clustering and the semi-supervised version for classifying. Both versions will be described using an unified framework. This algorithm can be shown to be an instance of the Classification EM (CEM) algorithm [5, 15] in the particular case of a gaussian mixture whose component densities have equal covariance matrices (section 3.3). In order to show that, we will make use of some basic results on linear regression and Bayes decision, they are introduced in section 3.2. For our application, we are interested in two class classification, we thus restrict our analysis to the two class case.

3.1 Theoretical Framework

We consider a binary decision problem where there are available a set of labeled data D_l and a set of unlabelled data D_u. D_u will always be non empty, whereas for unsupervised learning, D_l is empty.

Formally we will note, $D_l=\{(x_i, t_i)|i=1,\ldots,n\}$ where $x_i \in IR^d$, $t_i=(t_{1i},t_{2i})$ is the indicator vector for x_i and $D_u=\{x_i \mid i= n+1,\ldots,n+m\}$. The latter are assumed to have been drawn from a mixture of densities with two components C_1, C_2 in some unknown proportions π_1,π_2. We will consider that unlabeled data have an associated missing indicator vector $t_i=(t_{1i},t_{2i})$, $(i=n+1, \ldots, n+m)$ which is a class or cluster indicator vector.

3.2 Discriminant Functions

We give below some basic results on the equivalence between Bayes decision functions and linear regression that will be used for the interpretation of our learning algorithm as a CEM method.

3.2.1 Bayesian Decision Rule for Normal Populations

For two normal populations with a common covariance matrix $N(\mu_1,\Sigma)$ and $N(\mu_2,\Sigma)$ the optimal Bayesian discriminant function is [7]:

$$g_B(x) = (\mu_1-\mu_2)^t.\Sigma^{-1}.x + x_0 \tag{1}$$

Where x_0 is a given threshold. The decision rule is to decide C_1 if $g_B(x)>0$ and C_2 otherwise.

3.2.2 Linear Regression

Let X be a matrix whose i^{th} row is the vector x_i and Y be the corresponding vector of targets whose i^{th} element is a if $x_i \in C_1$ and b if $x_i \in C_2$. For a and b chosen such that $|C_1|.a + |C_2|.b = 0$, e.g. $a=\frac{|C_1|+|C_2|}{|C_1|}$ and $b= -\frac{|C_1|+|C_2|}{|C_2|}$, the solution to the minimization of the mean squared error (MSE) $\left\|Y - W^t X\right\|^2$ is:

$$W = \alpha.\Sigma^{-1}.(m_1 - m_2) \tag{2}$$

The corresponding discriminant function is :

$$g_R(x) = W^t (x-m) = \alpha.(m_1 - m_2)^t.\Sigma^{-1}.(x - m) \tag{3}$$

where m_k and Σ respectively denote the mean and the variance of the data for the partition C_k and α is a constant (see e.g. [7]). and m is the mean of all of the samples. The decision rule is: decide C_1 if $g_R(x)>0$ and otherwise decide C_2.

By replacing the mean and covariance matrix in (1) with their plug in estimate used in (2), the two decision rules g_B and g_R are similar up to a threshold. The threshold estimate of the optimal Bayes decision rule can be easily computed from the data so that regression estimate could be used for implementing the optimal rule if needed. For practical applications however, there is no warranty that the optimal Bayesian rule will give better results.

3.3 Classification Maximum Likelihood Approach and Classification EM Algorithm

In this section we will introduce the classification maximum likelihood (CML) approach to clustering [28]. In this unsupervised approach there are N samples generated via a mixture density:

$$f(x,\Theta) = \sum_{k=1}^{c} \pi_k f_k(x,\theta_k) \tag{4}$$

Where the f_k are parametric densities with unknown parameters θ_k, c is the number of mixture components, π_k is the mixture proportion. The goal here is to cluster the samples into c components P_1, .., P_c. Under the mixture sampling scheme, samples x_i are taken from the mixture density (4), and the CML criterion is [5, 15]:

$$\log L_{CML}(P,\pi,\theta) = \sum_{k=1}^{c} \sum_{i=1}^{N} t_{ki} \log\{\pi_k . f_k(x_i,\theta_k)\} \tag{5}$$

Note that this is different from the mixture maximum likelihood (MML) approach where we want to optimize the following criterion:

$$\log L_M(P,\pi,\theta) = \sum_{k=1}^{N} \log(\sum_{i=1}^{c} \pi_k . f_k(x_i,\theta_k)) \tag{6}$$

In the MML approach, the goal is to model the data distribution, whereas in the CML approach, we are more interested into clustering the data. For CML the mixture indicator t_{ki} for a given data x_i is treated as an unknown parameter and corresponds to a hard decision on the mixture component identity. Many clustering algorithms are particular cases of CML [5, 24]. Note that CML directly provides a partition of the data, for MML a partition can be obtained by assigning x to the group with maximal posterior probability $p(P_i/x)$.

The classification EM algorithm (CEM) [5, 15] is an iterative technique, which has been proposed for maximizing (5), it is similar to the classical EM except for an additional ***C***-step where each x_i is assigned to one and only one component of the mixture. The algorithm is briefly described below.

CEM

Initialization : start from an initial partition $P^{(0)}$

j^{th} *iteration*, $j \geq 0$:

E –step. Estimate the posterior probability that x_i belongs to P_k (i=*1,..., N*; k=1,…,c):

$$E[t_{ki}^{(j)} / x_i; P^{(j)}, \pi^{(j)}, \theta^{(j)}] = \frac{\pi_k^{(j)} . f_k(x_i;\theta_k^{(j)})}{\sum_{k=1}^{c} \pi_k^{(j)} . f_k(x_i;\theta_k^{(j)})} \tag{7}$$

C – step. Assign each x_i to the cluster $P_k^{(j+1)}$ with maximal a posteriori probability according to (7)

M–step. Estimate the new parameters ($\pi^{(j+1)}$, $\theta^{(j+1)}$) which maximize log $L_{CML}(P^{(j+1)}, \pi^{(j)}, \theta^{(j)})$.

CML can be easily modified to handle both *labeled and unlabeled* data, the only difference is that in (7) the t_{ki} for labeled data are known, (5) becomes:

$$\log L_C(P,\pi,\theta) = \sum_{k=1}^{c} \sum_{x_i \in P_k} \log\{\pi_k . f_k(x_i,\theta_k)\} + \sum_{k=1}^{c} \sum_{i=n+1}^{n+m} t_{ki} \log\{\pi_k . f_k(x_i,\theta_k)\} \quad (8)$$

CEM can also be adapted to the case of semi supervised learning: for maximizing (8), the t_{ki} for the labeled data are kept fixed and are estimated as in the classical CEM (E and C steps) for the unlabeled data.

3.4 CEM and Linear Regression

We will show now that CEM could be implemented using a regression approach instead of the classical density estimation approach. In CEM, parameters ($\pi^{(j)}$, $\theta^{(j)}$) are used to compute the posterior probability so as to assign data to the different clusters in the C-step. However, instead of estimating these probabilities, one could use a regression approach for directly assigning data to the different clusters during the C-step.

We will show that in the case of two normal populations with equal covariance matrices, these two approaches are equivalent.

For a given partition $P^{(j)}$ corresponding to an iteration of the CEM algorithm, suppose we perform a regression of the input matrix X whose columns are the x_i against the target vector Y whose i^{th} row is a if $x_i \in P_1^{(j)}$ and b if $x_i \in P_2^{(j)}$ with a and b as described in section 3.2.2. In this case, the decision rule inferred from the regression estimation together with an appropriate threshold, will be the optimal Bayes decision rule with plug in maximum likelihood estimates. Using this decision rule derived from the linear regression, we could then assign the data to the different clusters and the partition $P^{(j+1)}$ will be exactly the same as the one obtained from the classical mixture density version of CEM.

Therefore the E-step in the CEM algorithm may be replaced by a regression step and the decision obtained for the C-step will be unchanged.

Because we are interested here only in classification or clustering, if we use the regression approach, the M-step is no more necessary, the regression CEM algorithm could start from an initial partition, the j^{th} step consists in classifying unlabeled data according to the decision rule inferred from the regression. It can easily be proved that this EM algorithm converges to a local maximum of the likelihood function (5) for

unsupervised training and (8) for semi-supervised training. The algorithm is summarized below:

Regression-CEM

Initialisation : start from an initial partition $P^{(0)}$

j^{th} iteration, $j \geq 0$:

E-step : compute $W^{(j)} = \alpha.\hat{\Sigma}^{(j)^{-1}}.(m_1^{(j)} - m_2^{(j)})$

C-step : classify the x_is according to the sign$(W^{(j)}x_i + w_0^{(j)})$ into $P_1^{(j+1)}$ or $P_2^{(j+1)}$.

In the above algorithm, posterior probabilities are directly estimated via a discriminant approach. All other approaches we know of for unsupervised or semi supervised learning rely on generative models and density estimation. This is an original aspect of our method and we believe that this may have important consequences. In practice for classification, direct discriminant approaches are usually far more efficient and more robust than density estimation approaches and allow for example to reach the same performance by making use of fewer labeled data. A more attractive reason for applying this technique is that for non linearly separable data, more sophisticated regression based classifiers such as non linear Neural Networks or non linear Support Vector Machines may be used instead of the linear classifier proposed above. Of course, for such cases, the theoretical equivalence with the optimal decision rule is lost.

Regression CEM can be used both for unsupervised and semi supervised learning. For the former, the whole partition is re-estimated at each iteration, for the latter, targets of labeled data are kept fixed during all iterations and only unlabelled data are reassigned at each iteration. In the unsupervised case, the results will heavily depend on the initial partition of the data.

For our text summarization application we have performed experiments for both cases.

4 Automatic Text Summary System

4.1 A Base Line System for Sentence Classification

Many systems for sentence extraction have been proposed which use similarity measures between text spans (sentences or paragraphs) and queries, e.g. [8, 13]. Representative sentences are then selected by comparing the sentence score for a given document to a preset threshold. The main difference between these systems is the representation of textual information and the similarity measures they are using. Usually, statistical and/or linguistic characteristics are used in order to encode the text (sentences and queries) into a fixed size vector and simple similarities (e.g. cosine) are then computed.

We will build here on the work of [10] who used such a technique for the extraction of sentences relevant to a given query. They use a *tf-idf* representation and compute the similarity between sentence s_k and query q as:

$$Sim_1(q,s_k) = \sum_{w_i \in s_k, q} tf(w_i,q).tf(w_i,s_k).\left(1 - \frac{\log(df(w_i)+1)}{\log(n+1)}\right)^2 \quad (9)$$

Where, *tf(w,x)* is the frequency of term w in x (q or s_k), *df(w)* is the document frequency of term w and n is the total number of documents in the collection. Sentence s_k and query q are pre-processed by removing stop-words and performing Porter-reduction on the remaining words. For each document a threshold is then estimated from data for selecting the most relevant sentences.

Our approach for the sentence extraction step is a variation of the above method where the query is enriched before computing the similarity. Since queries and sentences may be very short, this allows computing more meaningful similarities. Query expansion - via user feedback or via pseudo relevance feedback - has been successfully used for years in Information Retrieval (IR) e.g. [30]. The query expansion proceeds in two steps: first the query is expanded via a similarity thesaurus - WordNet in our experiments - second, relevant sentences are extracted from the document and the most frequent words in these sentences are included into the query. This process can be iterated. The similarity we consider is then:

$$Sim_2(q,s_k) = \sum_{w_i \in s_k, q} \bar{tf}(w_i,q).tf(w_i,s_k).\left(1 - \frac{\log(df(w_i)+1)}{\log(n+1)}\right) \quad (10)$$

Where, $\bar{tf}(w,q)$ is the number of terms within the "semantic" class of w_i in the query q. This extraction system will be used as a baseline system for evaluating the impact of learning throughout the paper. Although it is basic, similar systems have been shown to perform well for sentence extraction based text summarization. For example [31] uses such an approach, which operates only on word frequencies for sentence extraction in the context of generic summaries, and shows that it compares well with human based sentence extraction.

4.2 Learning

We propose below a technique, which takes into account the coherence of the whole set of relevant sentences for the summaries and allows to significantly increasing the quality of extracted sentences.

4.2.1 Features

We define new features in order to train our system for sentence classification. A sentence is considered as a sequence of terms, each of them being characterized by a set of features. The sentence representation will then be the corresponding sequence of these features.

We used four values for characterizing each term w of sentence s: *tf*(w,s), $\bar{tf}(w,q)$, (1-(log(*df*(w)+1)/log(n+1)) and $Sim_2(q,s)$ -computed as in (10)- the similarity between q and s. The first three variables are frequency statistics which give the importance of a term for characterizing respectively the sentence, the query and the document. The

last one gives the importance of the sentence containing w for the summary and is used in place of the term importance since it is difficult to provide a meaningful measure for isolated terms [10].

4.2.2 The Learning Text Summary System

In order to provide an initial partition $P^{(0)}$, for the semi-supervised learning we have labeled 10% of sentences in the training set using the news-wire summaries as the correct set of sentences. And for the unsupervised learning we have used the baseline system's decision. We then train a linear classifier with a sigmoïd output function to label all the sentences from the training set, and iterate according to algorithm ***regression-CEM***.

5 Experiments

5.1 Data Base

A corpus of documents with the corresponding summaries is required for the evaluation. We have used the Reuters data set consisting of news-wire summaries [20]: this corpus is composed of 1000 documents and their associated extracted sentence summaries. The data set was split into a training and a test set. Since the evaluation is performed for a generic summarization task, collecting the most frequent words in the training set generated a query. Statistics about the data set collection and summaries are shown in table 1.

5.2 Results

Evaluation issues of summarization systems have been the object of several attempts, many of them being carried within the tipster program [21] and the Summac competition [27].

Table 1. Characteristics of the Reuters data set and of the corresponding summaries.

Collection	Training	Test	All
# of docs	300	700	1000
Average # of sentences/doc	26.18	22.29	23.46
Min sentence/doc	7	5	5
Max sentence/doc	87	88	88
News-wire summaries			
Average # of sentences /sum	4.94	4.01	4.3
% of summaries including 1st sentence of docs	63.3	73.5	70.6

This is a complex issue and many different aspects have to be considered simultaneously in order to evaluate and compare different summarizers [17].

Our methods provide a set of relevant document sentences. Taking all the selected sentences, we can build an *extract* for the document. For the evaluation, we compared this extract with the news-wire summary and used Precision and Recall measures, defined as follows:

$$\text{Precision} = \frac{\#\text{of sentences extracted by the system which are in the news - wire summaries}}{\text{total \# of sentences extracted by the system}}$$
$$\text{Recall} = \frac{\#\text{of sentences extracted by the system which are in the news - wire summaries}}{\text{total \# of sentences in the news - wire summaries}} \quad (11)$$

We give below the average precision (table 2) for the different systems and the precision/recall curves (figure 1). The baseline system gives bottom line performance, which allows evaluating the contribution of our training strategy. In order to provide an upper bound of the expected performances, we have also trained a classifier in a fully supervised way, by labeling all the training set sentences using the news-wire summaries.

Unsupervised and Semi-supervised learning provides a clear increase of performances (up to 9 %). If we compare these results to fully supervised learning, which is also 9% better, we can infer that with 10% of labeled data, we have been able to extract from the unlabeled data half of the information needed for this "optimal" classification.

Table 2. Comparison between the baseline system and different learning schemes, using linear sigmoid classifier. Performances are on the test set.

	Precision (%)	Total Average (%)
Baseline system	54,94	56,33
Supervised learning	72,68	74,06
Semi-Supervised learning	63,94	65,32
Unsupervised learning	63,53	64,92

We have also compared the linear Neural Network model to a linear SVM model in the case of unsupervised learning as shown at Table 3. The two models performed similarly, both are linear classifiers although their training criterion is slightly different.

Table 3. Comparison between two different linear models: Neural Networks and SVM in the case of Self-supervised learning. Performances are on the test set.

	Precision (%)	Total Average (%)
Self-Supervised learning with Neural-Networks	63,53	64,92
Self-Supervised learning with SVM	62,15	63,55

11-point precision recall curves allow a more precise evaluation of the system behavior. Let For the test set, let M be the total number of sentences extracted by the system as relevant (correctly or incorrectly), N_s the total number of sentences extracted by the system which are in the newswire summaries, N_g the total number of sentences in newswire summaries and N_t the total number of sentences in the test set.

Precision and recall are computed respectively as N_s/M and N_s/N_g. For a given document, sentence s is ranked according to the decision of the classifier. Precision and recall are computed for $M = 1,..,N_t$ and plotted here one against the other as an 11 point curve. The curves illustrate the same behavior as table 2, semi-supervised and unsupervised behave similarly and for all recall values their performance increase is half that of the fully supervised system. Unsupervised learning appears as a very promising technique since no labeling is required at all. Note that this method could be applied as well and exactly in the same way for query based summaries.

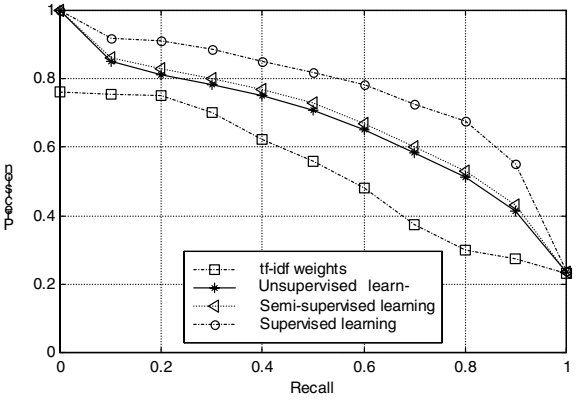

Fig. 1. Precision-Recall curves for base line system (square), unsupervised learning (star), semi-supervised learning (triangle) and the supervised learning (circle).

6 Conclusion

We have described a text summarization system in the context of sentence based extraction summaries. The main idea proposed here is the development of a fully automatic summarization system using a unsupervised and semi-supervised learning paradigm. This has been implemented using simple linear classifiers, experiments on Reuters news-wire have shown a clear performance increase. Unsupervised learning allows to reach half of the performance increase allowed by a fully supervised system, and is much more realistic for applications. It can also be used in exactly the same way for query based summaries.

References

1. Anderson J.A., Richardson S.C. Logistic Discrimination and Bias correction in maximum likelihood estimation. *Technometrics*, 21 (1979) 71-78.
2. Barzilay R., Elhadad M. Using lexical chains for text summarization. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, (1997) 10-17.
3. Blum A., Mitchell T. Combining Labeled and Unlabeled Data with Co-Training. Proceedings of the 1998 Conference on Computational Learning Theory. (1998).
4. Carbonell J.G., Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st ACM SIGIR, (1998) 335-336.
5. Celeux G., Govaert G. A classification EM algorithm for clustering and two stochastic versions. Computational Statistics & Data Analysis. 14 (1992) 315-332.
6. Chuang W.T., Yang J. Extracting sentence segments for text summarization: a machine learning approach. Proceedings of the 23rd ACM SIGIR. (2000) 152-159.
7. Duda R. O., Hart P. T. Pattern Recognition and Scene Analysis. Edn. Wiley (1973).
8. Goldstein J., Kantrowitz M., Mittal V., Carbonell J. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. Proceedings of the 22nd ACM SIGIR (1999) 121-127.
9. Klavans J.L., Shaw J. Lexical semantics in summarization. Proceedings of the First Annual Workshop of the IFIP working Group for NLP and KR. (1995).
10. Knaus D., Mittendorf E., Schauble P., Sheridan P. Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System. in TREC-4 proceedings. (1994).
11. Kupiec J., Pedersen J., Chen F. A. Trainable Document Summarizer. Proceedings of the 18th ACM SIGIR. (1995) 68-73.
12. Luhn P.H. Automatic creation of literature abstracts. IBM Journal (1958) 159-165.
13. Mani I., Bloedorn E. Machine Learning of Generic and User-Focused Summarization. Proceedings of the Fifteenth National Conference on AI. (1998) 821-826.
14. Marcu D. From discourse structures to text summaries. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. (1997) 82-88.
15. McLachlan G.J. Discriminant Analysis and Statistical Pattern Recognition. Edn. John Wiley & Sons, New-York (1992).
16. Miller D., Uyar H. A Mixture of Experts classifier with learning based on both labeled and unlabeled data. Advances in Neural Information Processing Systems. 9 (1996) 571-577.
17. Mittal V., Kantrowitz M., Goldstein J., Carbonell J. Selecting Text Spans for Document Summaries: Heuristics and Metrics. Proceedings of the 6th National Conference on AI. (1999).
18. Mitra M., Singhal A., Buckley C. Automatic Text Summarization by Paragraph Extraction. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. (1997) 31-36.
19. Nigam K., McCallum A., Thrun A., Mitchell T. Text Classification from labeled and unlabeled documents using EM. In proceedings of National Conference on Artificial Intelligence. (1998).
20. http://boardwatch.internet .com/mag/95/oct/bwm9.html
21. NIST. TIPSTER Information-Retrieval Text Research Collection on CD-ROM. National Institute of Standards and Technology, Gaithersburg, Maryland. (1993).
22. Radev D., McKeown K. Generating natural language summaries from multiple online sources. Computational Linguistics. (1998).
23. Roth V., Steinhage V. Nonlinear Discriminant Analysis using Kernel Functions. Advances in Neural Information Processing Systems. 12 (1999).

24. Scott A.J., Symons M.J. Clustering Methods based on Likelihood Ratio Criteria. Biometrics. 27 (1991) 387-397.
25. Sparck Jones K.: Discourse modeling for automatic summarizing. Technical Report 29D, Computer laboratory, university of Cambridge. (1993).
26. Strzalkowski T., Wang J., Wise B. A robust practical text summarization system. Proceedings of the Fifteenth National Conference on AI. (1998) 26-30.
27. SUMMAC. TIPSTER Text Summarization Evaluation Conference (SUMMAC). *http://www-nlpir.nist.gov/related_projects/tipster_summac/*
28. Symons M.J. Clustering Criteria and Multivariate Normal Mixture. Biometrics. 37 (1981) 35-43.
29. Teufel S., Moens M. Sentence Extraction as a Classification Task. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization. (1997). 58-65.
30. Xu J., Croft W.B. Query Expansion Using Local and Global Document Analysis. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1996). 4--11.
31. Zechner K.: Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. COLING. (1996) 986-989.