The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization

Massih-Reza Amini and Patrick Gallinari
Computer Science Laboratory of Paris 6 (LIP6)
University of Pierre and Marie Curie
4, place de jussieu, case 169
75252 Paris cedex 05, France
{amini, gallinari}@poleia.lip6.fr

ABSTRACT

With the huge amount of information available electronically, there is an increasing demand for automatic text summarization systems. The use of machine learning techniques for this task allows one to adapt summaries to the user needs and to the corpus characteristics. These desirable properties have motivated an increasing amount of work in this field over the last few years. Most approaches attempt to generate summaries by extracting sentence segments and adopt the supervised learning paradigm which requires to label documents at the text span level. This is a costly process, which puts strong limitations on the applicability of these methods. We investigate here the use of semi-supervised algorithms for summarization. These techniques make use of few labeled data together with a larger amount of unlabeled data. We propose new semi-supervised algorithms for training classification models for text summarization. We analyze their performances on two data sets - the Reuters news-wire corpus and the Computation and Language (cmp_lg) collection of TIPSTER SUMMAC. We perform comparisons with a baseline – non learning – system, and a reference trainable summarizer system.

Categories & Subject Descriptors: I.5.4 *text processing*, B.2.4 *Algorithms*, G.3. *Stochastic processes*.

General Terms: Algorithms, performance, design.

Keywords: text summarization, machine learning, semi-supervised learning, text-span extraction.

1. INTROUCTION

With the continuing growth of online text resources, it is becoming more and more important to help users to access information and to develop easy to use information research tools. Text summarization can be used together with conventional information research engines, and help users to quickly evaluate the relevance of documents or to navigate through a corpus.

Automated summarization dates back to the fifties [16]. 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 [30]. Simpler approaches were then explored which consist in extracting representative text-spans, using statistical techniques and/or techniques based on surface domain-independent linguistic analyses.

Within this context, 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 document sentences and selecting those with higher score and minimum overlap [6, 25]. Most of the recent work in summarization uses this paradigm. Usually, sentences are used as text-span units but paragraphs have also been considered [21, 31]. The latter may sometimes appear more appealing since they contain more contextual information. The quality of an *extract summary* might not be as good as an *abstract summary*, but it is considered good enough for a reader to understand the main ideas of a document.

Our work takes the text-span extraction paradigm and explores a machine learning approach for improving automatic summarization methods. The proposed model could be used both for generic and query-based summaries. However for evaluation purposes we will present results only on a generic summarization task. Previous work on the application of machine learning techniques for summarization [4, 8, 15, 17, 33] rely on the supervised learning paradigm. This requires a training set of documents and associated extract summaries. Learning systems are first trained to label document sentences as relevant when they are in the summary, or irrelevant otherwise. After training, they operate on unlabeled text by ranking the sentences of a new document. Labeling large amount of text spans for training summarization systems is time consuming and unrealistic for many applications. We consider here the use of semi-supervised techniques, which allow to train a system with only a few labeled documents together with large amounts of unlabeled documents. The latter being widely available and cheap, this could considerably help the development of trainable text summarizers.

For this, we introduce a new semi-supervised algorithm. Its originality is that it relies on a discriminative approach to semi-supervised learning rather than a generative approach, as it is usually the case. The advantage is that the algorithm is generic and can be used with many discriminant classifier, it leads to cheap and efficient implementations and shows better performances than generative systems. The algorithm is described in the framework of the Classification Expectation Maximization algorithm (CEM) [7, 19] and detailed for the case of a logistic classifier.

The paper is organized as follows, we first make a brief review of recent work in machine learning for text summarization and semi-supervised techniques (section 2). In section 3, we introduce two baseline text summarizers, a non trainable system and the Kupiec et al.'s trainable model [15], which are later used for comparison. We then describe our semi-supervised approach to text summarization

SIGIR'02, August 11-15, 2002, Tampere, Finland.

based on sentence extraction and present the formal framework of the model and its interpretation as a CEM instance (section 4). Finally we present a series of experiments on Reuters news-wire and on the Computation and Language (cmp_lg) of TIPSTER SUMMAC collections in section 6, and carry on a set of comparisons.

2. RELATED WORK

Recently, several innovative methods for automated document summarization have been explored, they exploit statistical approaches [6, 25, 31, 35], linguistic approaches [13, 18, 26] and combinations of both [3, 10].

We will focus here on a statistical approach to the problem and more precisely on the use of machine learning techniques, which has recently motivated an increasing amount of interest in the field. Trainable text summarizers allow one to adapt to the specific user needs and to corpus characteristics, they may also help to improve the quality of summaries.

Text extracting for summarization has been cast in the framework of supervised learning for the first time in the seminal work of [15]. The authors propose a generic summarization model, which is based on a naive Bayes classifier operating on a synthetic representation of sentences. This method is described in more detail in section 3.2. Different authors built on this idea, e.g. [33] apply the same technique for other training and evaluation settings. [17] have used several machine learning techniques in order to select features indicating the salience of a sentence. They considered three types of features (locational, thematic and cohesion) and addressed the production of generic and user-focused summaries. [8] compare three supervised learning algorithms: C4.5, naive Bayes and neural networks. Their conclusion is that all three methods successfully completed the task by generating reasonable summaries. [4] adopt supervised learning methods to make synthetic summaries of web pages in the Ocelot system.

All these approaches rely on the supervised learning paradigm and require the labeling of text spans as relevant or not relevant which is either performed manually or by aligning an abstract with document sentences. Manual tagging or abstracting is a tedious task, it is unrealistic for large corpora, for query based summaries or for adapting summaries to different user needs or corpora.

The semi-supervised learning paradigm has emerged as a solution to this type of problem when large corpora of unlabeled data are available together with a much smaller amount of labeled data. In our case, for a given document collection, labeling a few documents at the text span level, or producing a few abstracts is usually affordable and does not take much time.

From a machine learning perspective, automatic summarization by extracting is typically a task for which semi-supervised learning seems appropriate. This learning paradigm has been first explored in statistics, a review of the work done prior to 92 in the context of discriminant analysis may be found in [19]. Most approaches propose to adapt the Expectation Maximization (EM) algorithm for handling both labeled and unlabeled data and 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. Recently this paradigm has been rediscovered by the machine learning community and is subject to a growing interest. For example [20] adapt EM to a mixture of experts model, [23] propose an algorithm which is a particular case of the general semi-supervised EM described in [19], they extend it to multiple mixtures and present an empirical evaluation for text classification. [28] propose a Kernel Discriminant Analysis which can be used for semi-supervised classification.

Other ideas bear some similarity with semi-supervised learning. This is the case of the co-training paradigm [5] which has been proposed independently for training classifiers when data may be described with two modalities. The algorithm we propose here is related to the decision directed paradigm [9], which has been used under different settings in the field of adaptive signal processing.

On the machine learning side, the originality of our work lies in the design of a discriminative approach to semi-supervised learning whereas others mainly rely on generative classifiers. The main benefits are the following: the approach is fully generic in the sense that most discriminant classifiers could be used within this framework, it does not rely on any parametric assumption about the data, discriminative training allows to have better performances than generative methods especially when there are few training data, it leads to very simple and fast implementations.

3. AUTOMATIC SUMMARIZERS

In this section, we present two baseline systems for sentence extraction. The first system is a non-trainable statistical model (section 3.1) and the second is Kupiec et al.'s naive Bayes trainable system [15] (section 3.2).

3.1 A baseline non-trainable system

Many systems for sentence extraction rely on the use of similarity measures between text spans (sentences or paragraphs) and queries, e.g. [10, 17]. Representative sentences are then selected by comparing the sentence score for a given document to a preset threshold. These systems differ in the representation of textual information and in the similarity measures they use. 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 build here on the work of [14] 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:

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

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.

In our experiments we have considered a generic query, where the query is enriched before computing the similarity. Since queries and sentences may be very short, this allows to compute more meaningful scores. Query expansion proceeds in two steps: first the query is expanded via a similarity thesaurus - WordNet in our experiments - second, highly ranked sentences, according to the generic query, are extracted from the document and the most frequent words in these sentences are included into the query. This process can be iterated. We also take into account the sentence length when computing the similarity *tf*(*w*,*s*). The use of the document length has been shown to improve performance in Information Retrieval [27]. This leads to the following similarity:

$$Sim_2(q,s_k) = \sum_{w_i \in s_k, q} \bar{tf}(w_i,q).\bar{tf}(w_i,s_k).\left(1 - \frac{\log(df(w_i)+1)}{\log(n+1)}\right)^2$$

Where, $\bar{tf}(w,q)$ is the number of terms within the "semantic" class of w_i in the query q, and

$$\bar{tf}(w_i,s_k) = \frac{tf(w_i,s_k)}{tf(w_i,s_k) + \frac{\psi.l(s_k)}{\sum_s l(s)}}$$

Where ψ is the number of sentences in the current document, $l(s_k)$ is the length of s_k and $\sum_s l(s)$ is the summation over all sentence length.

Similar systems have been shown to perform well for sentence extraction based text summarization. For example [35] 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.

3.2 Naive Bayes Model

Our baseline trainable summarizer is the model proposed by Kupiec et al. [15]. Sentences are encoded into five discrete features $\{F_j\}_{j=1,..,5}$. Four of them are binary: a sentence length cut-of feature indicates whether or not the sentence length is below a specified threshold, a fixed-phrase feature is set to 1 if the sentence contains occurrences of cue words, thematic word feature indexes sentences whose similarity with a generic query is above a preset threshold (we have used the Sim_2 measure in our implementation), upper case word feature indicates the occurrence of acronyms, excluding common abbreviations. Finally a paragraph feature is set to 1, 2 or 3 depending on the position of the sentence in the text. The computation of the posterior probability that sentence *s* will be included in summary *S* is achieved assuming statistical independence of the features and using a naive Bayes classifier:

$$p(s \in S / F_1,..,F_5) = \frac{\prod_{j=1}^{5} p(F_j / s \in S).p(s \in S)}{\prod_{j=1}^{5} p(F_j)}$$

where $p(s \in S)$ is a constant, $p(F_j / s \in S)$ and $p(F_j)$ are estimated directly from the training set by counting occurrence. For computing $p(F_j / s \in S)$ and $p(F_j)$, we have used a smoothing scheme (where counts in both the numerator and denominator are augmented by one for each feature) to prevent zero probabilities for infrequently occurring features. The use of this type of prior is sometimes referred to as *Laplace smoothing*. In the paper we refer to this system as the naive Bayes classifier.

4. A Semi-supervised algorithm for text-span classification

We now introduce an iterative discriminant algorithm for semi-supervised learning applied to text-span extraction. This algorithm is generic in the sense that it can be used with any discriminant classifier provided its output can be interpreted as a posterior class probability. We describe our algorithm in the general framework of the Classification EM (CEM) algorithm [7, 19]. This ensures that all nice properties of CEM (e.g. convergence) hold for our method. For text summarization, we are interested in two-class classification (relevant or irrelevant for the summary), we thus restrict our presentation and analysis to this case. For simplifying further the presentation, we consider here only the case of logistic classifiers, which have been used in our experiments. These two hypotheses are not restrictive since the algorithm and analysis can be easily extended for any discriminant classifier and for multi-class problems.

We first introduce the CEM unsupervised algorithm and propose a semi-supervised version of this method. We then describe our algorithm in the particular case of logistic regression.

4.1 Framework

We consider a binary decision problem and suppose available a set of *m* unlabeled data D_u and a set of *n* labeled data D_l. We will denote, $D_u=\{x_i \mid i=n+1,\ldots,n+m\}$ and $D_l=\{(x_i, t_i) \mid i=1,\ldots,n\}$ where $x_i \in \mathbb{R}^d$, $t_i=(t_{1i},t_{2i})$ is the class indicator vector for x_i – here (t_{1i},t_{2i}) = (1,0) when sentence *i* is relevant and (0,1) otherwise. Data in D_u are assumed drawn from a mixture of densities with two components C_1, C_2 in some unknown proportions π_1 and π_2. We will consider that unlabeled data have an associated missing indicator vector $t_i = (t_{1i},t_{2i})$ for ($i=n+1$, …, $n+m$) which is a class or cluster indicator vector. The algorithms we consider attempt to iteratively partition the data into the two components C_1 and C_2. We also denote $(P_1^{(j)}, P_2^{(j)})$ the partition into two clusters computed by an algorithm at iteration *j*.

4.2 Classification Maximum Likelihood approach

The classification maximum likelihood (CML) approach [32] is a general framework which encompasses many clustering algorithms [7, 29]. It is only concerned with unsupervised learning, but we will see later that it can be easily adapted to semi-supervised learning.

In the two component case considered here, samples are supposed to be generated via a mixture density:

$$f(x,\Theta) = \pi_1.f_1(x,\theta_1) + \pi_2.f_2(x,\theta_2)$$

Where the f_k are parametric densities with unknown parameters θ_k and π_k is the mixture proportion. The goal here is to cluster the samples into 2 components P_1 and P_2. Under the mixture sampling scheme, samples x_i are taken from the mixture density *f* and the CML criterion is [7, 19]:

$$\log L_{CML}(P,\pi,\theta) = \sum_{k=1}^{2} \sum_{x_i \in D_u} t_{ki} \log\{\pi_k.f_k(x_i,\theta_k)\}$$

Where, $\sum_{x_i \in D_u}$ is a summation over all unlabeled samples which belong to D_u. This is different from the classical mixture maximum likelihood (MML) criterion. MML measures how well a model fits a data distribution, whereas CML measures the quality of the clustering performed with the model. For CML the mixture indicator t_{ki} associated to x_i is treated as an unknown parameter and corresponds to a hard decision about the membership of x_i to the k^{th} mixture component. It has to be learned together with the density parameters. Such a parameter is not explicitly present in the MML criterion.

The classification EM algorithm (CEM) [7, 19] is an iterative technique, which has been proposed for maximizing L_{CML}, 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 (For all x_i in D_u and $k\in \{1,2\}$):

$$E[t_{ki}^{(j)} / x_i; P^{(j)}, \pi^{(j)}, \theta^{(j)}] = \frac{\pi_k^{(j)}.f_k(x_i;\theta_k^{(j)})}{\sum_{k=1}^{2}\pi_k^{(j)}.f_k(x_i;\theta_k^{(j)})}$$

C – step. Assign each x_i to the cluster $P_k^{(j+1)}$ with maximal posterior probability according to $E[t/x]$

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

4.3 Semi-supervised generative-CEM

CML criterion can be easily extended for semi-supervised learning. Since the t_{ki} for labeled data are known, this parameter is either 0 or 1 for data in D_l. The new criterion – denoted here L_C - becomes:

$$\log L_C = \sum_{k=1}^{2}\left\{\sum_{x_i\in P_k}\log\{\pi_k.f_k(x_i,\theta_k)\} + \sum_{i=n+1}^{n+m} t_{ki}\log\{\pi_k.f_k(x_i,\theta_k)\}\right\}$$

The first summation inside the brackets is over the labeled samples, and the second one over unlabeled samples.

CEM can then be easily adapted for semi-supervised learning so as to maximize L_C instead of L_{CML}: for unlabeled data the t_{ki} are estimated as in the classical CEM (E and C steps), for labeled data, they are fixed to their known value. This is a *generative* approach to semi-supervised learning since it relies on the estimation of the f_k which model the data generation process. In this context, different models could be used depending on the distributional assumption about the input data. For our experiments the $\{f_k\}_{k=1,2}$ are assumed to be normal distributions.

After convergence, the model can be used for classification on new data using Bayes decision rule, i.e. x is assigned to the class with maximum a posteriori according to $p(P_k/x)$= E$[t_k/x]$.

4.4 Semi-supervised logistic-CEM

Because density estimation could be problematic especially for high dimensions or when only few data are labeled – this is what we are interested in - and since we are dealing with a classification problem, a more natural approach is to directly estimate the posteriors $p(P_k/x)$. This is known as the discriminant approach to classification.

For simplifying, we will consider here only the case of logistic classification [1]. In this case, the only distributional assumption is that the log likelihood ratio of class distributions is linear in the observations (see equation below), this is the case for a large family of distributions such as the exponential density family, (e.g. normal, beta, gamma, etc), and there is no explicit assumption on the nature of the densities.

$$\log\left(\frac{f_1(x)}{f_2(x)}\right) = \beta_0 + \beta^t.x, \text{where}\, \beta = (\beta_1,...,\beta_d)$$

With this model, posterior probabilities have the simple form of a logistic function:

$$p(P_1/x) = \frac{1}{1+\exp(-(\beta_0 + \beta^t.x))} \quad \text{and} \quad p(P_2/x) = 1 - p(P_1/x)$$

Let us now express $\log L_C$ as a function of the posteriors $p(P_k/x)$:

$$\log L_C = \sum_{k=1}^{2}\left\{\sum_{x_i\in P_k}\log\{p(P_k/x_i)\} + \sum_{i=n+1}^{n+m} t_{ki}\log\{p(P_k/x_i)\}\right\} + \sum_{i=1}^{n+m}\log\{f(x_i)\}$$

For discriminant classifiers, no assumption is made about the functional form of the marginal distribution $f(x)$, therefore, maximizing L_C is equivalent to maximizing $\tilde{L}_C$ [1]:

$$\log\tilde{L}_C(P,\beta_0,\beta) = \sum_{k=1}^{2}\left\{\sum_{x_i\in P_k}\log\{p(P_k/x_i)\} + \sum_{i=n+1}^{n+m} t_{ki}\log\{p(P_k/x_i)\}\right\}$$

Any discriminant classifier which estimates the posteriors $p(P_k/x)$ can be trained to optimize $\tilde{L}_C$. With the logistic regression model, one uses a simple logistic unit G whose parameters are (β_0, β), i.e. $G(x) = \frac{1}{1+\exp(-(\beta_0 + \beta^t.x))}$, $G(x)$ is used as an estimate of $p(P_1/x)$ and $1 - G(x)$ as an estimate of $p(P_2/x)$. The estimation of β's can be performed using the logistic-CEM algorithm described below:

Logistic-CEM

Initialization: Train a discriminant logistic model $G^{(0)}(x)$ over D_l, let $P^{(0)}$ be the initial partition obtained from this model on $D_l \cup D_u$.

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

E –step. Estimate the posterior probability that x_i belongs to P_k on D_u (i=n+1,…, n+m; k=1,2) using the output of the logistic classifier $G^{(j)}(x)$:

$$p(P_1^{(j)}/x) = \frac{1}{1+\exp(-(\beta_0^{(j)} + \beta^{t(j)}.x))} \quad \text{and} \quad p(P_2^{(j)}/x) = 1 - p(P_1^{(j)}/x)$$

C – step. Assign each $x_i \in D_u$ to the cluster $P_k^{(j+1)}$ with maximal posterior probability according to $p(P_k^{(j)}/x_i)$:

$$\forall x_i \in D_u, \forall k \in \{1,2\}, t_{ki}^{(j+1)} = sgn(p(P_k^{(j)} / x_i; \beta^{(j)}) - 0.5)$$

M–step. Find new parameters ($\beta_0^{(j+1)}$, $\beta^{(j+1)}$) which maximize log $\tilde{L}_C$ ($P^{(j+1)}$, $\beta_0^{(j)}$, $\beta^{(j)}$).

The algorithm is guaranteed to converge. In order to maximize $\tilde{L}_C$ in step M, we have used a quasi-Newton gradient procedure. The advantage of this method is that at each iteration it requires to compute only the first derivatives of $\log\tilde{L}_C$ with regard to the parameters β.

The main difference here with the generative method is that no assumption is made on the conditional densities f_1 and f_2 except that their log ratio is linear in x. The algorithm directly attempts to estimate the $p(P_k/x)$ - the quantity we are interested in - instead of the conditional densities. It will be shown later to outperform

[1] A similar argument has been developed for supervised learning in the case of logistic classifiers [1, 19].

significantly the generative approach, especially when there are few labeled data available. It does so by using fewer parameters than the generative approach and is faster to run.

The above algorithm can be used with any other discriminant classifiers which also estimate the posteriors. We have performed experiments using neural networks and support vector machines but they did not show any improvement over the simple logistic regression method described here, which in turn performed slightly better than a pure linear classifier.

5. DATA SETS

A corpus of documents with the corresponding summaries is required for the evaluation. We have used *a*) the Reuters data set consisting of news-wire summaries [11], this corpus is composed of 1000 documents and their associated extracted sentence summaries and b) the Computation and Language (cmp_lg) collection of TIPSTER SUMMAC [12]. This corpus is composed of 183 scientific articles. For the latter, we have used the text-span alignment method described by [2] to generate extract-based summaries from the abstract of each article in the collection. In this method, extractive summaries required for training are automatically generated as follows: the relevance of each document sentence with respect to the human summary is computed, and highest score sentences are retained for the reference extract.

In both cases, the data set was split into a training and a test set whose size was respectively 1/3 and 2/3 of the available data. The evaluation is performed for a generic summarization task, a query was generated by collecting the most frequent words in the training set.

6. EXPERIMENTS

6.1 Compression ratio

A compression ratio must be specified or computed for extractive summaries. Empirical tests on the Reuters data set show that the compression ratio (summary size / document size) decreases with the size of the document. Figure 1 plots this ratio as a function of the document length. The graph is roughly hyperbolic, this suggests that the summary length, which is the product of the compression and the document length, is approximately constant. This is in agreement with [10] who found that summary length is independent of document length on similar databases. For each document in the Reuters data set, it was then decided to extract the same number of sentences than in its corresponding news-wire summary.

For the cmp_lg collection we followed the SUMMAC evaluation by using 10% compression ratio [34].

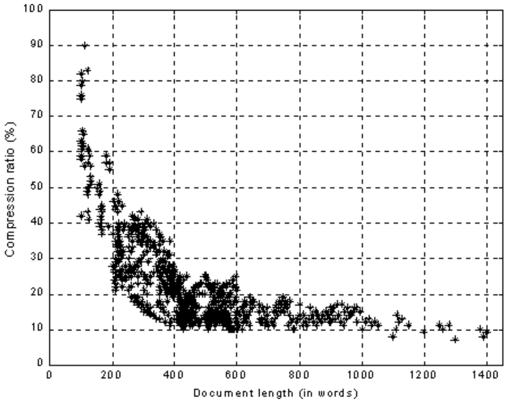

Figure 1. Compression as a percentage of document length on Reuters data set.

6.2 Sentence representation

We consider here the same set of features than those proposed by Kupiec et al. to represent sentences (section 3.2). The main difference here is that we do not use discrete feature values as they do and use both continuous and discrete features. This has been found more efficient and mixed continuous-categorical features are easily handled by our model. Each sentence *i* is then represented by a 5 feature vector, $\vec{x}_i$:

$$\vec{x}_i = (\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5)$$

φ_1 is the normalized sentence length: $\frac{l(i)}{\sum_j l(j)}$, φ_2 is the normalized frequency of cue words in sentence *i*: $\frac{frequency\, of\ cue\, words}{l(i)}$, φ_3 = $Sim_2(i, q)$ where Sim_2 is the similarity introduced in section 3.1 for the baseline non trainable system, φ_4 is the normalized frequency of acronyms in *i*: $\frac{frequency\, of\ acronyms}{l(i)}$ and φ_5 is the same paragraph feature as in [15].

6.3 Results

Evaluation issues of summarization systems have been the object of several attempts, many of them being carried within the tipster program [24] and the Summac competition. This is a complex issue and many different aspects have to be considered simultaneously in order to evaluate and compare different summarizers [22]. For the extraction task we are dealing with, things are a bit easier. We compared the extract of a system with the desired summary and used the following Precision and Recall measures:

$$\text{Precision} = \frac{\#\text{of sentences extracted by the system which are in the target summaries}}{\text{total \# of sentences extracted by the system}}$$

$$\text{Recall} = \frac{\#\text{of sentences extracted by the system which are in the target summaries}}{\text{total \# of sentences in the target summaries}}$$

We first compare the systems trained in a fully supervised way. This gives an upper bound of their performances and provides a first ranking of the different algorithms. Results are in table 1.

Table 1. Comparison between the baseline system and different learning classifiers on Reuters and cmp_lg test sets. All trainable systems are trained in a fully supervised way, using an expanded query. Accuracy is the ratio of correct classification for both relevant and irrelevant sentences.

	Reuters data set		Cmp_lg collection	
System	**Average Precision (%)**	**Accuracy (%)**	**Average Precision (%)**	**Accuracy (%)**
Baseline	53,23	55,13	54,53	56,16
Naive Bayes	61,02	63,03	61,83	63,48
Generative-CEM	72,86	73,06	74,12	74,79
Logistic-CEM	73,84	74,22	75,26	76,92

For both collections, naive Bayes is approximately 8% better in precision and accuracy than the non-trainable baseline and the CEM algorithms are about 11% better than naive Bayes. The logistic classifier being slightly better than the generative gaussian. Precision-Recall curves (figure 2 top) confirm this ranking. From bottom to top, the curves correspond respectively to a random sentence selection, to the first sentences of a document, to the baseline system, to naive Bayes fully supervised, to the logistic-CEM trained using 10% of the labels in the training set to generative and logistic fully supervised algorithms. The baseline system improves the break-even point performances respectively compared to random selection and first sentences by 15% and 6% on the Reuters data set and by 17% and 5,5% on the cmp_lg collection. Trainable systems clearly outperform the baseline system for all recall values and the two CEMs are clearly above the naive Bayes classifier. Obtaining similar results for both collections with training sets of respectively 330 and 60 documents suggests that the task is slightly easier for cmp_lg than for Reuters. This is confirmed by further experiments.

Another interesting result is that both logistic and generative-CEM trained with semi-supervised, using 10% of labeled documents together with 90% of unlabeled documents on the training set give similar performances than the naive Bayes classifier trained with all labeled documents on the training set. This suggests that our classifiers are sound enough to take advantage of a small a priori information, and that unlabeled data do indeed contain relevant information for this task.

Figures 2-bottom shows the average precision on both sets for semi-supervised learning at different ratio of labeled-unlabeled documents in the training set, for the generative and logistic semi-supervised algorithms. 10% on the *x*-axis, means that 10% of the labeled documents in the training sets were used for training, the 90% remaining being used as unlabeled training documents. For comparison, we have also performed test with a logistic classifier trained only on the labeled sentences without using the unlabeled sentences in the training set (logistic-supervised in fig. 2 bottom). Logistic-CEM uniformly outperforms all other systems. This is particularly clear for SUMMAC cmp-lg, which is a small document set. In this case, the discriminant approach is clearly superior to the generative approach. With only 10% of labeled documents in the training set, the logistic-CEM approach is over the baseline non trainable system and using about 15% of labeled documents allows to reach half the performance increase we can get with fully supervised training with regard to the baseline non trainable system. Using unlabeled sentences do increase the performances (compare e.g. logistic-CEM and logistic-supervised curves in figure 2-bottom at 10%), this confirms the soundness of semi-supervised learning for this task.

As for the complexity, logistic-CEM system uses only 6 β coefficients, one for each characteristic and one bias. Once trained, it is only needed to compute a weighted sum of the characteristics to rank sentences. The generative-CEM has 60 parameters since we used full gaussians (lower performances were obtained with diagonal covariance matrices here). Naive Bayes is approximately the same order of complexity than logistic. Logistic-CEM is then particularly cheap and fast to operate.

The algorithms described here are generic, and different versions using different classifiers may be implemented (any conditional density model can be used for the generative system and any discriminant classifier whose score approximates the posterior class probabilities can be used for the discriminant CEM). They can operate on any categorical and/or numerical representation of sentences or more generally of text spans, so that richer representations may lead to better performances. They can also be extended to handle directly *word sequences* instead of fixed size representations as it is described here. We have performed tests on simple numerical sequential representations of sentences. The performances were roughly the same as above, using only automatically computed frequential information. This opens the way to the selection of arbitrary text spans not corresponding to sentences or paragraphs. Such text spans could be used in more sophisticated abstracting systems. The system can also be used in a fully unsupervised way: a non-trainable baseline system is used to provide an initial labeling of the sentences, and the classifiers then learn to improve this labeling using a variant of the algorithms presented here. This *fully unsupervised* scheme provided results similar to the semi-supervised methods trained with 10 % labeled data and to the Kupiec et al. fully supervised naive Bayes method. However, the two datasets we have been using are clean and documents are well structured. The algorithms should be evaluated on more noisy data and the scaling of this system on larger collections with heterogeneous documents has still to be investigated. Since our methods are fully automatic and learn from the data, and also because different machine learning techniques can be used within this general framework, there is hope that the system degrades gracefully for more complex tasks, but this remains to explore.

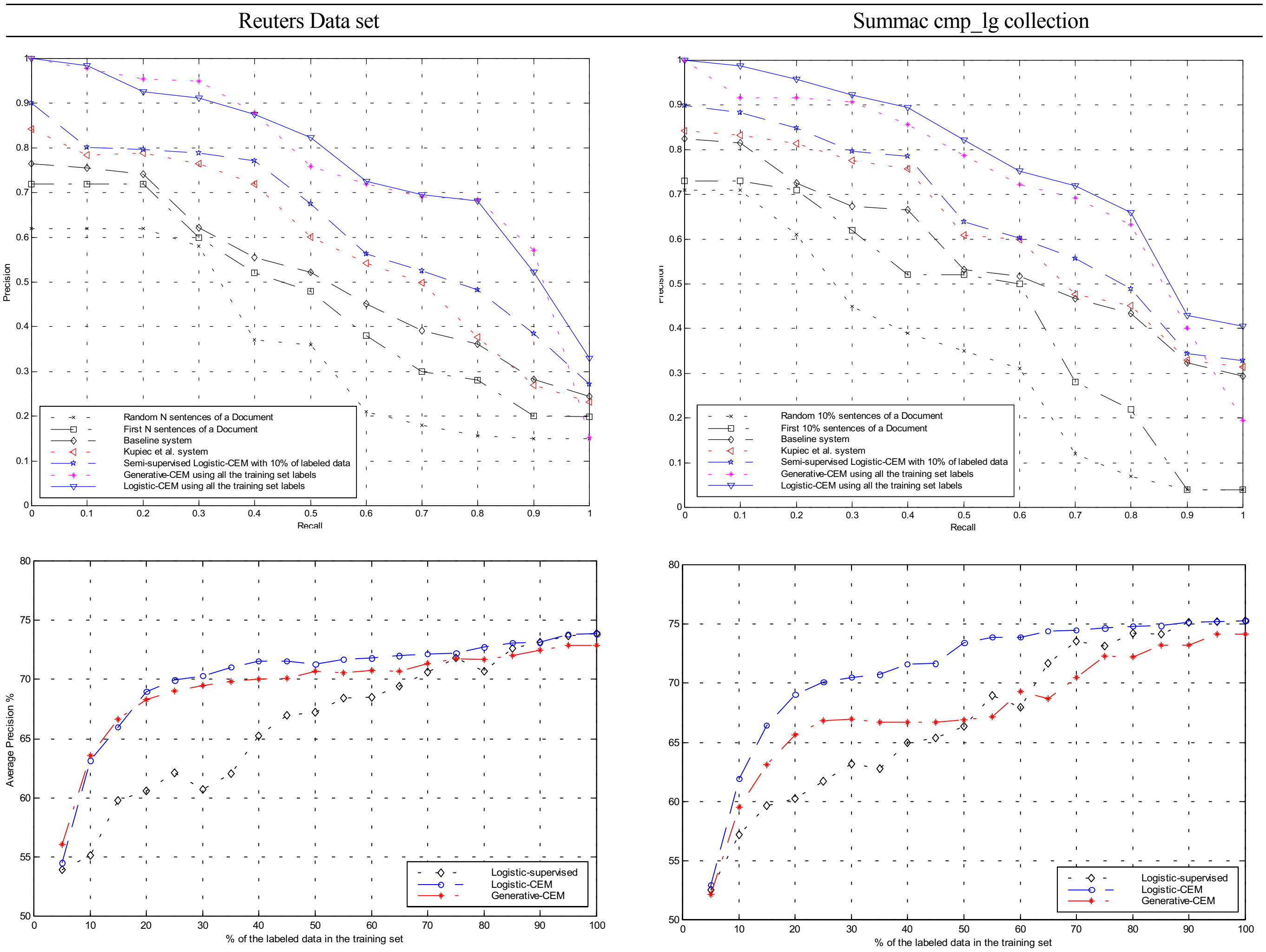

Figure 2. (Top) Precision recall curves for different systems. From bottom to top, the curves correspond respectively to a random selection of N first sentences, to the selection of N first sentences, to the baseline system, to Kupiec et al.'s system, to the CEM-logistic with 10% of labeled data and to generative and logistic fully supervised algorithms. (Bottom) Average precision of 3 trainable summarizers with respect to the ratio of labeled documents in the training set. The summarizers are the logistic and generative CEM algorithms and a logistic classifier trained only on *x*% labeled data.

7. Conclusion and future work

We have proposed a new general semi-supervised approach for training text summarizers based on sentence segment extraction and performed an evaluation on two data sets, the news-wire summaries of Reuters data and the cmp_lg collection of TIPSTER SUMMAC. Our method has been compared to a non-trainable baseline system and to the Kupiec et al.'s fully supervised learning classifier. With only 10% labeled documents, our algorithm reaches the performances of these systems and outperform them using more labeled data. Experiments show that using only 10 to 20% of labeled sentences in the training set allows to reach half of the performance increase provided by a fully supervised approach.

We have also compared discriminant and generative approaches to semi-supervised learning and the former has been found clearly superior to the latter for small collections.

Finally we have briefly mentioned extensions of this system to sequential text representations and to unsupervised learning, which shows that this system is really flexible.

In future work, we plan to evaluate our systems on large and heterogeneous data sets.

Acknowledgements

Many thanks to Vibu O. Mittal who helped us a lot to improve this work and proposed further directions to investigate.

8. REFERENCES

[1] Anderson J.A., Richardson S.C. Logistic Discrimination and Bias correction in maximum likelihood estimation. *Technometrics*, 21 (1979) 71-78.

[2] Banko, M.; Mittal, V.; Kantrowitz, M.; and Goldstein, J. Generating Extraction-Based Summaries from Hand-Written One by Text Alignment. Pac. Rim Conf. on Comp. (1999)

[3] 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.

[4] Berger A.L., Mittal V.O., OCELOT: A system for summarizing web pages. Research and Development in Information Retrieval (2000).

[5] Blum A., Mitchell T. Combining Labeled and Unlabeled Data with Co-Training. Proceedings of the 1998 Conference on Computational Learning Theory. (1998).

[6] 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.

[7] Celeux G., Govaert G. A classification EM algorithm for clustering and two stochastic versions. Computational Statistics & Data Analysis. 14 (1992) 315-332.

[8] Chuang W.T., Yang J. Extracting sentence segments for text summarization: a machine learning approach. Proceedings of the 23rd ACM SIGIR. (2000) 152-159.

[9] Duda R. O., Hart P. T. Pattern Recognition and Scene Analysis. Edn. Wiley (1973).

[10] 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.

[11] http://boardwatch.internet .com/mag/95/oct/bwm9.html

[12] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster_summac/cmp_lg.html

[13] 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).

[14] Knaus D., Mittendorf E., Schauble P., Sheridan P. Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System. in TREC-4 proceedings. (1994).

[15] Kupiec J., Pedersen J., Chen F. A. Trainable Document Summarizer. Proceedings of the 18th ACM SIGIR. (1995) 68-73.

[16] Luhn P.H. Automatic creation of literature abstracts. IBM Journal (1958) 159-165.

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

[18] Marcu D. From discourse structures to text summaries. Proceedings of the ACL’97/EACL’97 Workshop on Intelligent Scalable Text Summarization. (1997) 82-88.

[19] McLachlan G.J. Discriminant Analysis and Statistical Pattern Recognition. Edn. John Wiley & Sons, New-York (1992).

[20] 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.

[21] 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.

[22] 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).

[23] 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).

[24] NIST. TIPSTER Information-Retrieval Text Research Collection on CD-ROM. National Institute of Standards and Technology, Gaithersburg, Maryland. (1993).

[25] Nomoto T., Matsumoto Y., A new Approach to Unsupervised Text Summarization. Proceedings of the 24th ACM SIGIR (2001) 26-34.

[26] Radev D., McKeown K. Generating natural language summaries from multiple online sources. Computational Linguistics. (1998).

[27] Robertson S., Sparck-Jones K., Relevance weighting of search terms. Journal of the American Society for Information Science, 27 3, (1976) 129-146.

[28] Roth V., Steinhage V. Nonlinear Discriminant Analysis using Kernel Functions. Advances in Neural Information Processing Systems. 12 (1999).

[29] Scott A.J., Symons M.J. Clustering Methods based on Likelihood Ratio Criteria. Biometrics. 27 (1991) 387-397.

[30] Sparck Jones K.: Discourse modeling for automatic summarizing. Technical Report 29D, Computer laboratory, university of Cambridge. (1993).

[31] Strzalkowski T., Wang J., Wise B. A robust practical text summarization system. Proceedings of the Fifteenth National Conference on AI. (1998) 26-30.

[32] Symons M.J. Clustering Criteria and Multivariate Normal Mixture. Biometrics. 37 (1981) 35-43.

[33] 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.

[34] Tipster text phase III 18-month workshop notes, May 1998. FairFax, VA.

[35] Zechner K.: Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. COLING. (1996) 986-989.