Interactive Learning for Text Summarization

Massih-Reza AMINI
LIP6, University of Paris 6
Case 169, 4 Place Jussieu, F – 75252
Paris cedex 05, France.
amini@poleia.lip6.fr

Abstract

This paper describes a query-relevant text summary system based on interactive learning. The system proceeds in two steps, it first extracts the most relevant sentences of a document with regard to a user query using a classical tf-idf term weighting scheme, it then learns the user feedback in order to improve its performances. Learning operates at two levels: query expansion and sentence scoring.

1. INTRODUCTION

With the increase of textual information, it becomes important for information retrieval to provide relevant information quickly and in a suitable form. Document summaries are convenient for the user since they allow him to rapidly consult retrieved documents and to decide which of them are really in his field of interest.

Automated summarization dates back to the fifties [Luh 58]. Human-quality text summarization is considered too problematic since it encompasses discourse understanding, abstraction, and language generation [Spa 93]. A simpler approach consists in extracting representative *text-spans*, it avoids the central difficulties of natural language processing. Most recent work in summarization uses this paradigm. The text-span extraction paradigm transforms the problem of summarization into a simpler problem of ranking sentences from the original document according to their relevancy. *Generic summarization* consists in abstracting the main ideas of a whole document, whereas *query based summarization* aims to abstract the information relevant for a given query. Generally, sentences are used as text-span units but paragraphs have also been considered [Mit 97, Str 98]. The latter may sometimes appear more appealing since they contain more contextual information.

Most of the work on extraction-based summarization uses simple statistical measures on terms (e.g. term frequency) and ad hoc linguistic units for the text representation [Car 98, Gol 99]. Computing similarities between sentences then performs scoring.

Some authors have proposed to use machine learning for improving summarization systems. [Kup 95] and [Teu 97] consider the problem of sentence extraction as a statistical classification task. [Bar 97] explore the use of lexical chains. Like in statistical keyword based methods, they model document concepts by using content-specific lexical items observed in the text. [Jin 99] propose the use of HMM models to generate automatically human-written summary sentences, the model is trained upon human generated summaries.

We consider here the text summarization problem as a sentence extraction paradigm and we use user interaction to train our system. Our model could be used both for generic and query-based summaries, however its is better suited for the latter since it relies on human interaction which is not central for generic summarization tasks. For clarity, we will consider in the following only query-based summaries. Compared to previous works [Kup 95, Teu 97], the originality of our approach is that it does not require a corpus of document and associated summaries for training the system, although we use such a corpus for the evaluation and for simulating user interaction in this evaluation. Supervision is provided here by the user interaction. Another originality is that learning operates at two levels: query expansion and sentence scoring via a classification system.

The paper is organized as follows: we first introduce a baseline system that is used in a first step for sentence extraction (section 2). We then describe the learning step (section 3), and finally we present a series of experiments (section 4 and section 5).

2. SENTENCE EXTRACTION BY USING SIMILARITY MEASURES

The system proceeds in two steps, it first extracts the most relevant sentences of a document with regard to a user query using a classical tf-idf term weighting scheme, it then learns the user feedback in order to improve its performances. We describe below the first step.

Many systems for sentence extraction have been proposed which use similarity measures between text spans (sentences or paragraphs) and queries, e.g. [Gol 99, Mit 97]. 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 characteristic 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 [Kna 94] 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:

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

Where, *tf(w,d)* is the frequency of term *w* in *d*, *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 to compute more meaningful similarities. Query expansion appeared to be very important in our experiments.

For expanding the query, we proceed 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:

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

Where, $\bar{tf}(w,q)$ is the number of terms within the "semantic" class of w_i in the query q. Query expansion - via user feedback or via pseudo relevance feedback - has been successfully used for years in Information Retrieval (IR) e.g. [Xu 96, Gau 99].

This extraction system will be used as a baseline system for evaluating the impact of learning and user interaction throughout the paper. Although it is really basic, similar systems have been shown to perform well for sentence extraction based text summarization. For example [Zec 96] 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. Learning

Methods based on similarity measures do have intrinsic limitations: they rely on simple predefined features and measures, they are developed for generic documents, their adaptation to a specific corpus, to different document genres or to a specific community of users has to be manually settled.

Several authors have advocated the use of machine learning methods to improve the qualities or the adaptability of summarization systems [Gol 99, Jin 99, Kup 95, Teu 97]. This is particularly important in an Internet context since document types and user demands may vary considerably. Machine learning allows exploiting both corpus characteristics and user interaction.

We propose below a technique, which takes into account these two aspects, and allow to significantly increase the quality of the extracted sentences.

In the proposed approach, learning is based on user interaction. User's query is first processed by the baseline system of section 2 and the user is then presented with a summary consisting of the N most relevant sentences for each document. For each summary, sentences are ranked according to their relevance. The user can then give a feedback by selecting sentences he considers as relevant or irrelevant.

Learning based on this feedback will then operate at two levels:

- Feedback will be used to change the query representation as in classical relevance feedback, we used for that an approach similar to [Roc 71]. This allows to expand the query according to the user judgment and to refine the query expansion performed during the first step (section 2).

- A classifier is trained to classify sentences according to their relevance for the summary. Let R_Q be a binary variable, which is equal to 1 if a sentence is relevant for the summary and for query Q, and 0 if not. Relevant sentences are those selected by the baseline system which have not been changed into irrelevant by the user and vice versa for irrelevant sentences. The classifier will be trained to compute $P(R_Q /s)$ for each sentence s of a document. Using a classifier offers a principle way to compute the relevance of a sentence, it allows to adapt the decision function to the corpus and to the user needs whereas methods like the one in section 2 make use of a predefined metric which does not take into account these two aspects, it allows to automatically weight the features used for representing the terms in sentences.

In order to define more precisely the way this classifier operates we need to describe the features used for text representation and the training set.

Features

Our text features must be informative for the task of query based summarization. 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*: *tf*(*w,s*), $\bar{tf}(w,q)$, (1-(log(*df*(*w*)+1)/log(*n*+1)) and *Sim*(*q*,s) 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 [Kau 94].

Training set

In order to train our classifier we need a labeling of the document into summary-relevant and irrelevant sentences. This labeling is provided by the baseline system and the user feedback: sentences which have been selected by the baseline system will be considered relevant and others as irrelevant, the user can change the label of these sentences.

The classifier is then trained to compute $P(R_Q/s)$ using this training set.

Classifier

In order to make feasible the computation of $P(R_Q/s)$ we have to make some simplifying assumptions.

Let $w_1^n = w_1 ... w_n$ denote the sequence of feature vectors representing a sentence *s* of length *n*. In our case, each *w* is a vector of length 4.

Under the assumption that the feature vectors are independent ($P(w_1^n) = \prod_i P(w_i)$) and that $P(R_Q / w_1^i) = P(R_Q / w_i)$, it is easily shown that

$$P(R_Q / s) = \prod_{w_i \in s_k} P(R_Q / w_i) \qquad (1)$$

Equation (1) will be used for computing the score of a document.

The only justification for the above assumptions is that they make the computation of (1) feasible and provide a score for ranking the sentences. In practice, we have found more efficient to use the following decomposition:

$$P(R_Q / s) = \prod_{w_i \in s_k} P(R_Q / w_{i-2}^{i+2}) \qquad (2)$$

Which can be justified using a slight modification of the above assumptions. In (2), we consider the local context of term w_i (a window of size 2 around w_i) for computing its relevancy score, whereas in (1) this context is ignored. The window size has been set experimentally.

For the implementation of this classifier, we have used a multi-layer perceptron [Bis 95] since this is among the most efficient classifier systems.

4. DATABASE

A corpus of documents with the corresponding summaries is required for the evaluation. Note that as already said such a corpus is not necessary for implementing the proposed system, but it allows here to simulate user interaction (see below) and to evaluate the system performances. We have used the Reuters data set consisting of news-wire summaries[1]: 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. The training set was used to simulate a query consisting of the most frequent words in this set, and the corresponding summaries were used to learn the system's parameters. Statistics about the data set collection and summaries are shown in table 1.

Reuters data set			
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

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

Figure 1-a shows that the histogram of summary length in sentences is narrowly distributed around 5 sentences and Figure 1-b highlights that the summary length in words is approximately a normal distribution with a peak around 80 words.

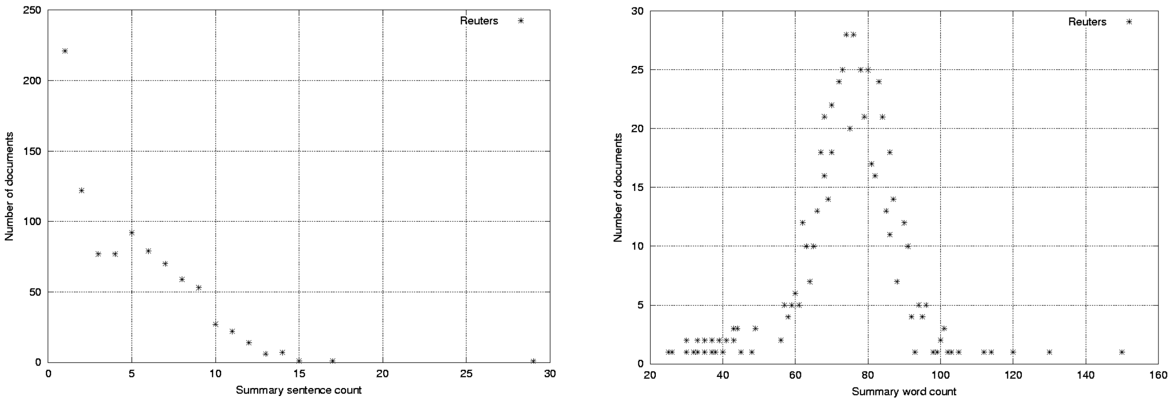

Figure 1. a: Distribution of summary sentence length, b: Distribution of summary word length.

5. EVALUATION

Evaluation issues of summarization systems have been the object of several attempts among which extensive one have been the tipster program [NIST 93] and the Summac competition [SUM 98]. This is a complex issue and many different aspects have to be considered simultaneously in order to evaluate and compare different summarizers [Mitt 99].

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

We will consider here classical measures commonly used in the domain, they are the precision / recall curves and the F-ratio:

$$\mathrm{Pr}\,ecision = \frac{\#of\ sentences\ extracted\ by\ the\ system\ \ which\ are\ \ in\ the\ news - wire\ summaries}{total\ \#of\ \ sentences\ extracted\ by\ the\ system}$$

$$Recall = \frac{\#of\ \ sentences\ extracted\ by\ the\ system\ \ which\ are\ in\ the\ news - wire\ summaries}{total\ \#of\ \ sentences\ in\ the\ news - wire\ summaries}$$

$$F - measure = \frac{2 \times \mathrm{Pr}\,ecision \times Recall}{Recall + \mathrm{Pr}\,ecision}$$

The sentences extracted by the system were compared with the sentences in the news-wire summaries. We achieved an average 53,94% precision, 53,96% recall and 53,95% F-measure with the baseline system and up to an average 72,68% precision, 72,71% recall, and 72,69% F-measure when using feedback and learning. This is a significant (about 20 %) performance increase which shows the importance of feedback and the soundness of our classification scheme.

For quantifying the importance of feedback, we have performed different tests by varying the degree of feedback. For simulating the interaction with the user, we considered a subset (denoted here the feedback-subset) of the training set and for each document in this subset, we changed the label of the sentences selected by the baseline system, if they were not present in the newswire document summary.

Table 2 shows the performances of the system on the test set when the feedback-subset consists of 10 % of the training set and of the whole training set.

	Precision (%)	Recall (%)	F-Mesure (%)
Baseline system	54,94	53,96	53,95
Learning on 10% of the training set	63,94	63,96	63,95
Learning on the whole training set	72,68	72,71	72,69

Table 2. Comparison between the baseline system and the learning system for two different feedback-subsets. Performances are on the test set.

System precision is increased by more than 10% if only 10% of the training set summaries are used to provide feedback. It goes up another 10 % when we use the whole training set for feedback.

Figure 2 shows this behavior in more details for different sizes of the feedback set. The performances gradually increase with the degree of feedback. For comparison, we have also plotted the performances of the system when only the classifier is trained without query expansion. The performances are lower, they increase up to a plateau and then slightly degrade due to overfitting.

11-point precision recall curves allow to evaluate more precisely the system's behavior. Let M be the total number of sentences extracted by the system as relevant (correct or incorrect), 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 the total number of sentences in test set.

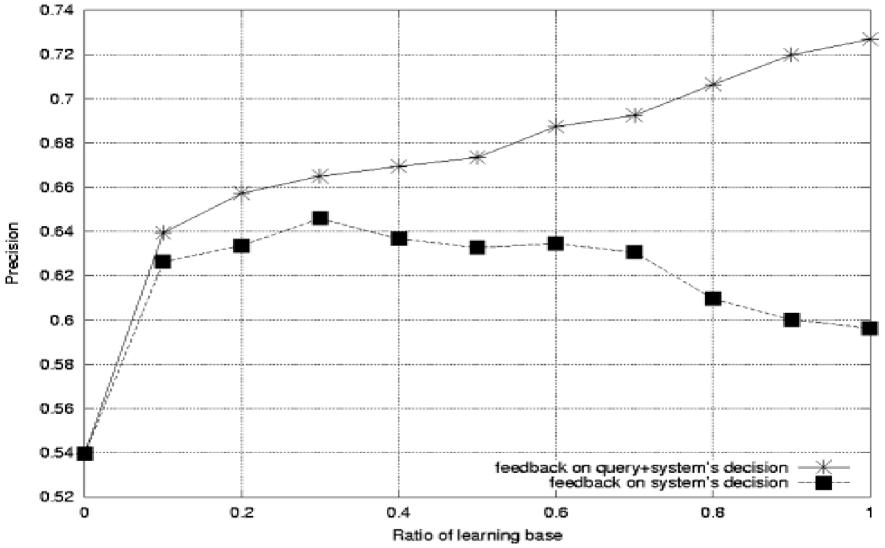

Figure 2. System's precision as a function of the degree of feedback in the training set. The *x*-axis is the ratio of documents in the training set, which are used for feedback (feedback-subset size).

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 $P(R_Q/\ s)$. The top *M* are classified as relevant and the rest as irrelevant. Precision and recall are computed for M = 1,..,N and plotted here one against the other as an 11 point curve.

Figure 3 compares the recall precision curves for the baseline system and for the learning system. Two curves for learning are computed for a feedback-subset of a) 10% of the training set and b) the whole training set. The interaction over 10% of the training set already leads to increase appreciably the performances of the learning system.

Figure 4 shows the evolution of the precision-recall curves as a function of the ratio of the training set used for interaction with the learning system. We notice that the system performances mainly increase in the high-recall, low-precision region space.

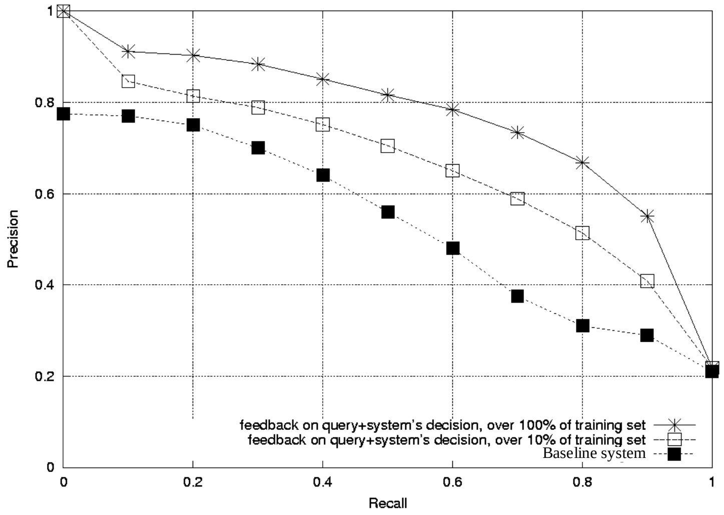

Figure 3. Precision recall curves for the baseline system and the learning system.

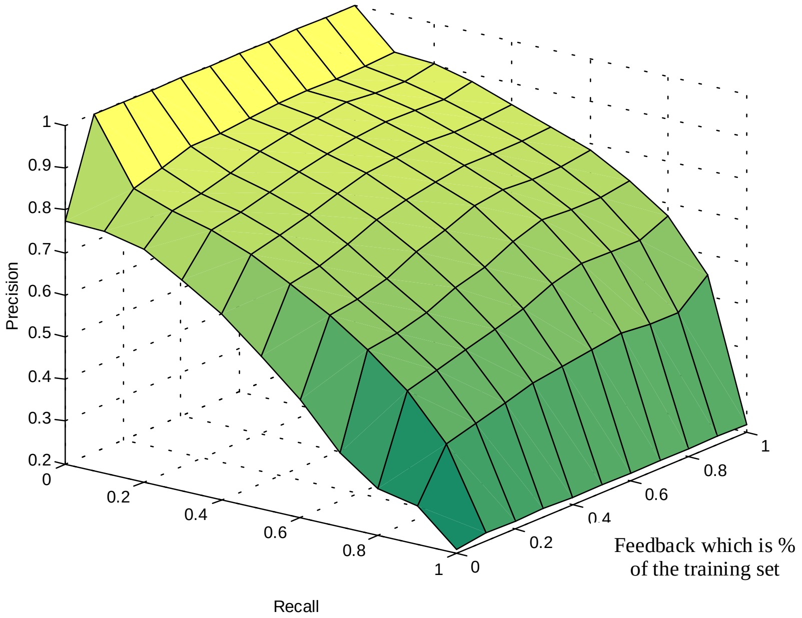

Figure 4. The precision-recall curves as a function of the feedback-subset size expressed in % of the training set.

6. CONCLUSION

We have proposed a text summarization system in the context of sentence based extraction summaries. This system learns upon user feedback by modifying the queries and by training a classifier to rank document sentences according to their relevance for the query. Learning allows to adapt the summarizer to the corpus and to user needs. Our system does not need a document - summary corpus, but only user feedback in order to learn. We feel that this is an important issue since the development of summaries is a long and tedious task, and that there are only a few document - summary corpus.

Our experiments show that learning based on feedback can increase significantly the performances of a classical baseline system. We are currently investigating the use of automatic feedback and semi supervised learning in order to provide fully automatic summaries, when there is no user feedback or when this feedback is very limited.

7. BIBLIOGRAPHY

[Bar 97] Barzilay R., Elhadad M. « Using lexical chains for text summarization » *In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL*, pp.10-17.

[Bis 95] Bishop C. « Neural Networks for Pattern Recognition », *Book*, Oxford University Press, 1995.

[Car98] Carbonell J.G., Goldstein J., « The use of MMR, diversity-based reranking for reordering documents and producing summaries », *Proceedings of SIGIR'98*, 1998, Melbourne, Australia.

[Gau 99] Gauch S., Wang J., Rachakonda S.-M. « A Corpus Analysis Approach for Automatic Query Expansion and its Extension to Multiple DataBases», *ACM Transactions on Information Systems*, Vol. 17, No. 3, pp. 250--269, 1999.

[Gol 99] Goldstein J., Kantrowitz M., Mittal V., Carbonell J., « Summarizing Text Documents: Sentence Selection and Evaluation Metrics », *Proceedings of SIGIR'99,* 1999.

[Jin 99] Jing H., McKeown K. «The decomposition of Human-Written Summary Sentences», *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99),* 1999, CA.

[Kna 94] Kanus D., Mittendorf E., Schauble P., Sheridan P., « Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System », *in TREC-4 proceedings.*

[Kup 95] Kupiec J., Pedersen J., Chen F. « A Trainable Document Summarizer », *In the proceedings of the 18th ACM SIGIR conference on research and development in information retrieval*, pp. 68-73, 1995.

[Luh 58] Luhn P.H. « Automatic creation of literature abstracts », *IBM Journal* (1958), pp.159-165.

[Mit 97] 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, Madrid, Spain.

[Mitt 99] Mittal V., Kantrowitz M., Goldstein J., Carbonell J., « Selecting Text Spans for Document Summaries: Heuristics and Metrics », *Proceedings of AAAI'99,* 1999, Orlando. USA.

[NIST 93] NIST, « TIPSTER Information-Retrieval Text Research Collection, on CD-ROM», *published by The National Institute of Standards and Technology*, Gaithersburg, Maryland, 1993.

[Roc 71] Rocchio J. « Relevance feedback in Information Retrieval », *Smart Retrieval System: Experiments in Automatic Document Processing, Book,* Chapter 14, pp. 313--323, Prentice-Hall, 1971.

[Spa 93] Sparck Jones K. « Discourse modeling for automatic summarizing », *Technical Report 29D,* Computer laboratory, university of Cambridge, 1993.

[Str 98] Strzalkowski T., Wang J., Wise B., « A robust practical text summarization system », *Proceedings of AAAI'98,* 1998, pp.26--30, Stanford.

[SUM 98] « TIPSTER Text Summarization Evaluation Conference (SUMMAC) », *http://www-nlpir.nist.gov/related_projects/tipster_summac/*

[Teu 97] Teufel S., Moens M. « Sentence Extraction as a Classification Task », *Workshop: Intelligent and scalable Text summarization, ACL/EACL 1997,* 1997.

[Xu 96] Xu J., Croft W.B. « Query Expansion Using Local and Global Document Analysis », *in Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 4--11, 1996.

[Zec 96] Zechner K. « Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences ». *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96),* pp. 986--989, Denmark.