Description of the LIPN Systems at TAC 2009

Aurélien Bossard
Laboratoire d'Informatique de Paris-Nord
CNRS UMR 7030 and Université Paris 13
93430 Villetaneuse — France
{abossard}@lipn.univ-paris13.fr

1 Introduction

The Text Analysis Conferences (TAC) offer a unique occasion to show innovative approaches to text summarization. As a first incursion into this new research area, LIPN participated in the Update Summarization task of TAC 2008. The LIPN wanted to improve the results obtained during TAC 2008 and to confirm that the changes made to its summarization system really enhanced the quality of the automatically created summaries.

This paper gives a technical description of the two systems developed for TAC 2009. Algorithms and results are then briefly discussed.

In the first section, we describe the LIPN1-Update system, which is based on our previous system – CBSEAS – ran for TAC 2008 Update and Opinion Summarization task. This system has been improved by the mean of a genetic algorithm for finding the parameters that best suit the task. We then describe our second system, LIPN2-Update, which integrates a co-reference analyzer.

2 LIPN1-Update

The system we used for TAC 2009 is CBSEAS, which is described in details in (Bossard, Généreux, and Poibeau 2009).

We assume that redundant pieces of information are the most important thing in order to produce a good summary. Therefore, the sentences which carry those pieces of information have to be extracted. Detecting groups of sentences conveying the same information is the first step of our approach. The developed algorithm first establishes the similarities between all sentences of the documents to summarize, then apply a clustering algorithm — fast global k-means (Lopez-Escobar, Carrasco-Ochoa, and Martinez Trinidad 2006) — to the similarity matrix in order to create clusters in which sentences convey the same information.

The system then extracts one sentence per cluster. The sentence extracted is the one that maximize the following parameters:

- proximity to the center of its class;
- similarity to user query or topic;
- similarity to user-defined desired sentence length;
- score based on sentence position in the document structure (described in (Bossard and Poibeau 2009)).

We only changed three aspects of our system for TAC 2009:

- the named entity recognition, used for sentence similarity computation and similarity to the user query, is now processed with a robust named entity tagger (Cunningham et al. 2002);
- sentence similarity measure is computed differently (cf 2.1);
- the different parameters are optimized using a genetic algorithm trained on past Update Task data.

2.1 New Sentence Similarity Measure

Each sentence is represented by different morpho-syntactic vectors. The similarity between two sentences consists in a weighted sum of the similarities of the two sentences vectors. This new similarity measure is shown in *fig. 1.*

The similarity between two terms is computed as follows :

- 1 if two terms are morphologically equal;
- using WordNet (Fellbaum 1998), the JCn similarity measure (Jiang and Conrath 1997) if not morphologically equal.

$$sim(p_1, p_2) = \frac{\sum_{tm \in T} weight(tm) \times fsim(s_1, s_2)}{\frac{fsim(s_1, s_2) + gsim(p_1, p_2)}{\sum_{tm \in T} weight(tm)}}$$

$$fsim(s_1, s_2) = \sum_{n_1 \in s_1} \sum_{n_2 \in s_2} tsim(n_1, n_2) \times \frac{tfidf(n_1) + tfidf(n_2)}{2}$$

$$gsim(s_1, s_2) = card\left((n_1 \in s_1, n_2 \in s_2) \mid tsim(n_1, n_2) < \delta\right)$$

T is the list of morpho-syntactic and named entity types, s_1 s_2 the sentences, and $tsim$ the similarity function between two terms, δ is a similarity threshold under which two terms are considered completely different.

Figure 1: Similarity measure used in CBSEAS

2.2 Sentences Filtering

The documents contain undesirable sentences. It is the case of endnotes which are frequent in the TAC 2008 data. More than one third of the documents contain an endnote. Most of them are links to external resources which provide additional content for the news subject. They are often closely related to the document title, and so to the topic query. They have a high probability of being extracted in the summary, but are not good candidates. They must be filtered before providing the documents to CBSEAS. We erased every sentence beginning by "On the Net", "On the Web", "See also".

We do not want CBSEAS to have too much sentences as input. This would disturb the clustering algorithm, create clusters of sentences which are not as close to the query as we would like, and cause our system to extract undesirable sentences. We limited the number of sentences that can be taken as input (we further explain in 2.5 how this number is set) and select the sentences which are closest to the user query.

2.3 Clustering Algorithm

We cluster the sentences using fast global k-means, an iterative version of k-means. This algorithm is easy to adapt, and this has proven to be of use when working on slightly different tasks than automatic summarization, such as "update task" of TAC campaign. Indeed, we slightly modified the algorithm to manage the update problematic so that the novelty is detected without having to set a similarity threshold between a new sentence and the sentences from the first document set above which the sentence is considered as conveying new information. This is explained in details in our paper for TAC 2008 ((Bossard, Généreux, and Poibeau 2009)).

2.4 Description of CBSEAS parameters

After having clustered the sentences, the system extracts one sentence per cluster. This is done in order to eliminate redundancy (all similar sentences should be in the same cluster) and to improve diversity in the summary. The sentence chosen is the one that maximizes the weighted sum of all the following scores (we explain in 2.5 how we set the weights) :

local centrality The local centrality reflects how central the sentence is in its cluster. The more central the sentence is, the more it reflects the overall information content of its cluster.

For each sentence, we compute the sum of the similarities to the other sentences of its cluster. The local centrality score is this sum normalized by the number of sentences in the cluster.

similarity to user query We have found during TAC 2008 that taking only in account the local centrality was not sufficient. In fact, some selected sentences were not giving an answer to the user query but were selected as they were central in their cluster.

To avoid that, we introduced another score, depending on the user query. For each sentence, we compute the similarity to the user query, using the same similarity measure as the one described in 2.1.

sentence length Long sentences often contain useless information. Long sentences also disturb the reader who is awaiting concise sentences when reading a summary. We also want to avoid too short sentences as we want to be as close as possible to the limit of 100 words. The length score is computed as follows :

$$length_{score} = log(|length(sentence) - length_{user}|)$$

sentence position We automatically classify news articles in five different types : chronologies, speech report, technical files, opinion report, and classical news. More information about those categories is available in (Bossard and Poibeau 2009). Chronologies and technical files are written in a very concise style, and sentences from those kind of articles are good candidates for being selected in the summary. We gave a bonus score to sentences from chronologies and technical files

if the similarity of their title to the user query is above a threshold. We also gave a bonus to the first three sentences of the news classified in one of the three other categories.

2.5 Genetic Algorithm

The weights used in our similarity measure and in the sentence selection phase were previously set empirically. For TAC 2009, we have trained the parameters using a genetic algorithm on a limited number of summary sets from TAC 2008 Update Task, using ROUGE SU4 as the fitness score. Computing a summary and its score is a time-costing task, so we only trained CBSEAS on 5 summary sets.

Here is the description of our genetic algorithm :

Individuals selection method The evaluation of one individual is for us a time costly operation. That is the reason why have chosen a tournament selection method, which has the advantage to be easily parallelized. For each generation of γ individuals, μ tournaments between λ individuals are organized.

The winner of each tournament is selected to be part of the next generation parents. Another advantage of this method lies in the fact that it preserves diversity because the selected individuals are not forced to be the best ones. This prevents the algorithm to fall in a local minimum.

Mutation operator As we don't know what parameters are dependant one to another, we want to change several parameters at the same time. In order to avoid a too heavy variation due to the simultaneous mutation of several parameters, we have chosen to limit the variation quantity of a parameter, weakening the probability to obtain a strong variation. We do that by using a logarithmic variation.

Creating a new generation Each generation is composed of 100 individuals. The algorithm organizes twenty tournaments with fifteen randomly selected representatives. This seems to be a good compromise between quick evolution and diversity preservation.

Each new generation is composed of the twenty winners, forty individuals created by mutating the winners, and the last forty created by randomly crossing the winners.

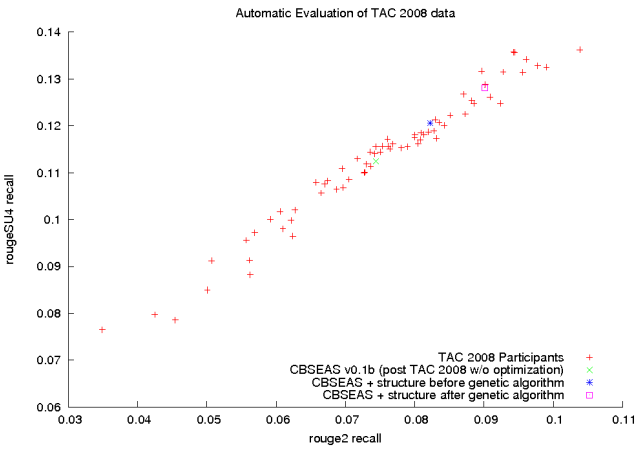

Figure 2: Results of our modified system on TAC 2008 data

2.6 LIPN-1 Update Evaluation on TAC 2008 data

We evaluated the changes made to our system by using the ROUGE automatic measures on TAC 2008 data. This evaluation, presented in figure 2 shows great improvements. This has encouraged us to evaluate our system on TAC 2009. The main interest of submitting our results to TAC 2009 evaluation is to get more detailed results and manual evaluation.

3 LIPN-2 Update

Terms can have multiple referents. It is important, in order to compute an accurate similarity measure between sentences or between a sentence and a query, to address the co-reference problem.

In order to be as readable as possible, a summary should not contain pronouns without their referent. A co-reference disambiguizer could also be used for that purpose.

We used the same system as the one used for LIPN-1 run, except that in the final summary, every pronoun has been replaced with its referent and all named entities have been replaced with their shortest co-referent. The ANNIE co-reference disambiguizer from the GATE platform (Cunningham et al. 2002) was used for co-reference resolution.

	Linguistic quality	Ov. resp.
LIPN-1 A	4.591	4.364
LIPN-2 A	4.136	4.023
LIPN-1 B	4.750	4.000
LIPN-2 B	4.659	3.977

Figure 3: LIPN-1 and LIPN-2 manual evaluation

4 Results and Discussion

LIPN1 LIPN-1 system results are good, as the system is ranking in the top quarter of all participants (cf *fig. 4*. The system seems to perform better on B summaries if looking at pyramid scores, but seems to be performing better on A summaries if looking at ROUGE-SU4 scores. However, pyramid scores should be given more importance, as it is a manual evaluation guided by a well-defined protocol (Nenkova, Passonneau, and McKeown 2007).

This tends to prove that our system manages the update summarization efficiently. However, the evaluation still does not take into account redundancy between A and B summaries. It is specified that B summaries shall not include information appearing in the first document set. The future evaluations have to take that spectification into consideration.

LIPN2 LIPN-2 system performance is disappointing (cf *fig. 4*). If lower ROUGE-SU4 scores can be explained by the deletion of pronouns from the summaries, that have been replaced with their referent, the also lower pyramid scores prove that the pronouns and the entities have not been replaced with the good referent. The co-reference matcher does not perform as well as it should.

The LIPN-2 system did perform worse on linguistic quality also (cf *fig. 3*). Having replaced the pronouns and entities with their referent does not only lower the informative score, but also affects the linguistic quality and the readability of the summaries. Although the named entity tagger obtained good results, the co-reference matcher doesn-t seem to be enough efficient to be integrated as is to our summarization system.

5 Conclusion

In this paper, we have presented the two systems that we used to participate in TAC 2009 Update task.

The LIPN systems ran for TAC 2009 Update Task

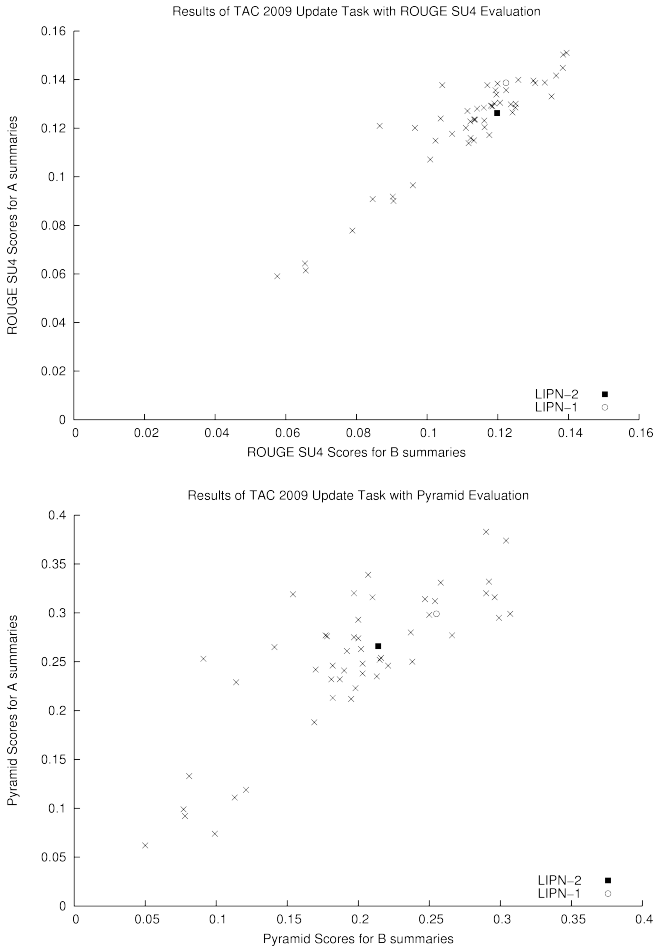

Figure 4: LIPN-1 and LIPN-2 Evaluation using ROUGE-SU4 and Pyramid

performed well. The co-reference matcher used in our system did not perform enough well to enhance the results. Its use had the opposite effect. In order to address the problem of co-reference resolution, we should look at other co-reference resolution systems that would better match the summarization task, that requires systems that favour precision.

Our systems can be improved in other ways, such as improving the score based on document structure (Bossard and Poibeau 2009).

References

Bossard, A., and Poibeau, T. 2009. Integrating document structure to an automatic summarizer. In *RANLP 2009*.

Bossard, A.; Généreux, M.; and Poibeau, T. 2009. De-

scription of the lipn systems at tac2008: Summarizing information and opinions. In *TAC 2008 Workshop*.

Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Jiang, J. J., and Conrath, D. W. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR* cmp-lg/9709008.

Lopez-Escobar, S.; Carrasco-Ochoa, J. A.; and Martinez Trinidad, J. F. 2006. Fast global k-means with similarity functions algorithm. In *IDEAL*, 512–521.

Nenkova, A.; Passonneau, R.; and McKeown, K. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.* 4(2):4.