The Italica System at TAC 2008 Opinion Summarization Task

Fermín Cruz, José A. Troyano, Javier Ortega, Fernando Enríquez
ITALICA
University of Seville
{fcruz,troyano,javierortega,fenros}@us.es

1 Introduction

The TAC 2008 Opinion Summarization Pilot Task consists on generating multidocument summaries with certain peculiarities. First, the documents to be processed come from blogs and can be considered *opinion* texts. Second, the goal is not to obtain a summary of the complete documents, but the answers given in the documents to several questions. As input to the system, apart from the documents themselves, the participants are provided with the questions and also the snippets that answer those questions, being the latter generated by the QA systems participating in another TAC 2008 task.

Our system is based on the combination of the snippets provided for the summary construction. In order to make the text more readable and complete the information, the process starts looking for the most relevant sentences in relation to the snippets, which are then used to generate the text that will finally compose the summary. Our system is therefore focused on the sentence extraction phase. We have obtained good results with the pyramid F-score and overall responsiveness measures, achieving the second and first place respectively among the participating systems.

In the following sections we describe the architecture of the system in first place, and continue discussing the results obtained in the evaluation process. Finally, we conclude discussing the strong and weak points of our system.

2 System Architecture

The system we have developed (figure 1) works in the following manner. Starting with the html documents provided, we generate plain text documents separated by sentences and conveniently tokenized. Given a question, we use the snippets to find the most relevant sentences in the related documents. The retrieved sentences are then introduced in a clause extraction phase, in which the decision of whether each of these sentences has to be used completely or if just one of its clauses is of our interest is taken. If the size of the set of sentences and clauses selected exceeds a certain threshold, the most significant sentences are selected by means of a clustering procedure and a sentence quality measure based on a regression tree. Finally, we make some simple transformations to the sentences, looking forward to increase the quality of the summary obtained. We are now proceeding to explain each one of these steps in more detail.

2.1 HTML documents pre-processing

We extract from the html documents all the text that would be visualized in a web browser, including the labels of the links. A big amount of the extracted text can be considered as noise, and must be filtered later on. We use the OpenNLP [5] tokenizer and sentence splitter. (We also use the OpenNLP chunker and parser tools in other parts of the process).

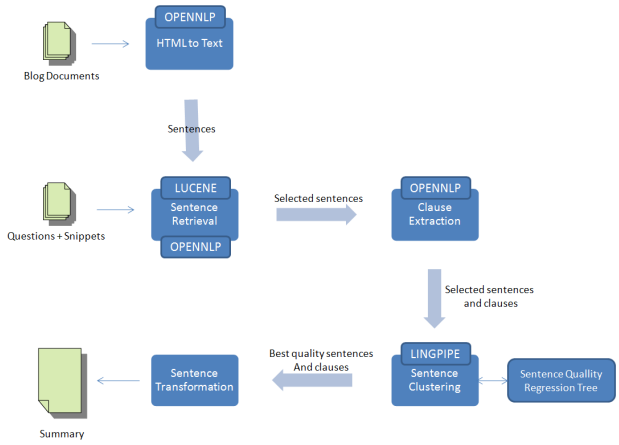

Figure 1: System architecture

2.2 Sentence retrieval

As many automatic summarization systems do ([3][8] among others), all the sentences we obtained in the previous stage are indexed using an information retrieval engine (we have used Lucene [7]). For each of the snippets, we search for the most relevant sentence in the document that it has been associated with. We take for granted that the snippets have been extracted (almost) completely literally from the documents. What we intend to do is to retrieve that sentence as a minimal unit to compose the summary. To retrieve the most relevant sentence, we follow a strategy based on a progressive decrease of demands. We begin literally searching for the snippet in the document. If we cannot find any sentence, we look for some other sentence containing all the phrases in the snippet. If we still have no results, we start eliminating phrases from the query, starting with the ones that have less number of words (the phrases with one word, two words, ...). If finally none of the searches produces a result, or the result obtained has a score given by Lucene lower than 0.75, the retrieval has failed. In this case, we use the snippet directly as the summary composition unit.

2.3 Clause extraction

For each of the sentences obtained in the retrieval phase, we look if it has any clause smaller than the whole sentence containing all the words of the corresponding snippet (excluding stop words). If we find such clause, we use it instead of using the complete sentence. This way, we pretend to minimize the inconsistencies in the discourse structure of the summary, maintaining the essential information.

2.4 Sentence clustering

Based on the sentence ranking step described in [2], and only in case we exceed the maximum size permitted for the summary (7000 non-whitespace characters per question), we make use of a sentence selection process, based on the redundancy. We use LingPipe [1] to apply a clustering process to the sentences, using the term coocurrence as a distance measure between them. For each of the generated clusters, we choose the sentence with better quality, employing for this a regression tree trained previously with manually labeled samples.

2.4.1 Sentence quality

Measuring the linguistic quality of a sentence is a previously used component in summarization systems [4] [6], but we are facing this problem in a different, supervised machine learning way. To measure the quality of a sentence, we have trained a regression tree with WEKA [9]. Two members of the work group labeled with a value in the range from 1 to 5 the linguistic quality of 84 sentences extracted from the snippets provided by the organization as examples. The purpose was to label negatively those sentences that we would prefer not to be part of the summary due to a matter of forms. The labels that presented a bigger divergence among the members involved were discussed by the team, and the mean of the two values assigned was used for the rest. Each of the sentences is transformed into a vector using features based on the syntax tree and morphological information.

The model obtained was hence used to select the better sentence among the components of a cluster

(in case the maximum number of characters permitted had been exceeded). Furthermore, in run number 2, we used a minimum quality score that all the sentences had to surpass in order to allow them to be part of the summary.

2.5 Sentence transformation

Once we have the sentences that constitute the summary, we carried out some simple transformations, some of them based on [6]:

- We add the dots at the end of sentences which are missing and we make sure that the initial letter in a sentence is a capital letter.
- We eliminate some usual phrases or pet words in opinion texts at the end of the sentences, like ", I think." or ", huh?".
- We undo some of the tokenization effects, e.g. erasing the spaces between personal pronouns and auxiliary verb contractions.
- We change some constructions from first to third person, e.g. "I think" is changed to "People think".

3 Evaluation

Our system has done fairly well with measures related with the information recall of the generated summaries. The two runs that have been evaluated are completely automatic, that is to say that we have not tuned any system parameter to adjust it for the test data. The only difference between them is that in run number two we force a minimum quality for the sentences to be part of the summary, using in that terms the value generated by the regression tree. Our two runs have achieved the second and third best results with the pyramid F-score, and the first and second best results in overall responsiveness (table 1). We believe that the good results of our summaries as for the relevant information recall is due to the fact that we have focused mainly on the retrieval of the most relevant sentences. However, in the overall fluency/readability we obtain a worse result (tenth place). Particularly, the worst result we obtain is in non-redundancy, what makes us suspicious of the needs of applying the clustering stage in all cases (now we use it only in those cases where we have obtained a summary that is too long as it exceeds the limits set by the organization).

	Run 1	pos	Run 2	pos
Pyramid F-score	0.490	2	0.489	3
Grammaticality	5.591	10	5.545	12
Non-redundancy	5.318	29	5.364	28
Structure/Coherence	3.273	9	2.682	19
Overall fluency/readability	3.909	10	3.591	19
Overall responsiveness	5.773	1	5.409	2

Table 1: ITALICA system evaluation

On the other hand, the second run obtains worse results with all measures. We believe the problem may be that the training data used for the construction of the model that measures the quality of the sentences are too few, thus the quality measures that we generate are not reliable.

4 Conclusions

In this paper we present our experience in participating in the TAC 2008 Opinion Summarization Task. Such task consists in the creation of summaries extracted from several blog documents responding to a group of questions. Our system takes as inputs the blogs, the questions and the answers given to those questions in snippets provided by the organization.

We present an extraction based approach, in which the selection of the sentences that better match the content of the snippets plays an important role. In this sentence selection module we have developed an adaptable information retrieval strategy that assures the obtaining of the nearest sentence to a given snippet. Once these sentences are found, the system is completed with a series of modules. The goal for these modules is to achieve improvements in the readability and correctness of the summary, by eliminating the elements in the sentences that are not relevant, eliminating the redundancy by means of clus-

tering, measuring the quality of the sentences or deploying simple syntax transformations.

The results of our system are favourable in general when they are related with the contents recall (pyramid F-score and overall responsiveness) and worse in readability and redundancy. We therefore consider that our sentence selection strategy is correct while the solutions used to improve the quality and readability of the summary are certainly insufficient.

References

[1] Alias-i. LingPipe 3.7.0. `http://alias-i.com/lingpipe`, 2008.

[2] Kirk Roberts Andrew Hickl and Finley Lacatusu. LCC's GISTexter at DUC 2007. Machine Reading for Update Summarization. In *Proceeding of the Document Understanding Conference 2007*, 2007.

[3] Yllias Chali and Shafiq R. Joty. University of Lethbridge's Participation in DUC-2007 Main Task. In *Proceeding of the Document Understanding Conference 2007*, 2007.

[4] Annibale Elia Tsvi Kuflik Ernesto, D'Avanzo and Simonetta Vietri. LAKE System at DUC-2007. In *Proceeding of the Document Understanding Conference 2007*, 2007.

[5] OpenNLP forum. Opennlp. `http://opennlp.sourceforge.net`, 2006.

[6] Michael Gamon Jagadeesh Jagarlamudi Hisami Suzuki Kristina Toutanova, Chris Brockett and Lucy Vanderwende. The PYTHY Summarization System. Microsoft Research at DUC 2007. In *Proceeding of the Document Understanding Conference 2007*, 2007.

[7] Lucene. The Lucene search engine. `http://lucene.apache.org`, 2005.

[8] Horacio Rodríguez Maria Fuentes and Daniel Ferrés. FEMsum at DUC 2007. In *Proceeding of the Document Understanding Conference 2007*, 2007.

[9] Weka Machine Learning Project. Weka. `http://www.cs.waikato.ac.nz/~ml/weka`.