Using SSM for Enhancing Summarization

Abdullah Bawakid and Mourad Oussalah

School of Engineering
Department of Electronic, Electrical and Computer Engineering
University of Birmingham
{ axb517 , M.oussalah }@bham.ac.uk

Abstract:

This paper describes a query-based multi-document summarizer that was built to participate in the summarization task of TAC11. The Similar to last year's system, it relies on a thesaurus extracted from Wikipedia and uses it as its underlying ontology. In addition, the system was updated by including a Sentences Simplification Module (SSM) that is applied in an iterative process in the post-processing stage. SSM also affects how sentences are ranked and chosen to form the summaries. The evaluation results and the performance of the system are provided.

1 Introduction

The Text Analysis Conference (TAC) is one of the well-known workshops in the field of Natural Language Processing which provides the infrastructure necessary to evaluate different methodologies with different tasks. In TAC11, we participated in the Guided Summarization task with two different runs. The aim of the task is to provide short summaries for a set of newswire articles. The generated summaries are not to exceed 100 words each. This year's task is similar to the one given last year in that the participants are asked to perform a deeper semantic analysis of the source documents instead of simply relying on documents words frequencies to select the important concepts. For this, a list of categories and important aspects for each category are given and it is asked that the summary provided should cover all of the mentioned aspects if possible in addition to any other information related to the topic. The test documents were taken from the newswire portion of the TAC 2010 KBP source data, rather than ACQUAINT and ACQUAINT-2 as was performed in the past year.

To enhance the representation of the documents to summarize in each set, the developed system described in this paper applies a set of rules to expand the document representation with the help of an external ontology. In our first participation in the Summarization task of TAC08, we relied on WordNet as an external ontology [1]. In TAC10, we used Wikipedia instead. In this year's task, we also use Wikipedia. However, we included a Sentences Simplification Module in the built system. The module aims to shorten the length of sentences via either splitting or compression. Details about the ontology we used along with a description of how it was built was reported in our earlier work [2].

The rest of this paper is structured as follows: We describe the built system and how it was applied to this year's task. Next, we present the evaluation results and discuss the rank, the strength and the limitations of our system. Finally, the paper is concluded with a potential future work.

2 System Overview

The system developed for the summarization task is extractive. Each sentence is assigned a score signifying its importance based on its extracted features. The summary is then generated for sets A by ranking the sentences based on their assigned scores in a descending order and choosing the top n sentences till the maximum word-limit is reached. Before adding a candidate sentence to the summary, an iterative process handling redundancy is applied. The stages involved for creating summaries are summarized in the following subsections:

2.1 Preprocessing

The first stage in the framework is to preprocess all fed documents by cleaning them and then parsing them to extract the text and topics and then tokenizing the terms and splitting the sentences. The stop words are then removed.

3.2 Identifying the concepts

We employed the built Wikipedia-thesaurus and its extracted features to detect concepts within documents through an exact match measure where explicitly mentioned concepts within each sentence are detected. A concept label having multiple spellings and synonyms should still be detected by the system as a single concept. This is due to the integration of redirect links within the thesaurus and the mapping algorithm that associates sentences with the concepts they contain. As for ambiguous terms and concepts, the system implements the Weighted Strong Links method that was described in [2].

3.3 Feature Selection

Each sentence is tagged with several features. These features are used to compute a score determining the sentence importance.

Overlap with the Topic: In our system, we consider the overlap between each sentence and the topic of its document set. We take into account both the concepts overlap and the terms overlap when assigning a score to each sentence. Synonyms and concepts with alternative spellings are considered as a single concept in our system with the help of the Wikipedia thesaurus and the custom matcher.

Concepts Dominance: The explicitly mentioned concepts within a document set which are most frequent and the topic concepts are considered to be the most important. When computing a score for each sentence based on this feature, we consider how pertinent the sentence concepts to the important concepts with the document set. We use the relevancy degrees between the concepts which are precomputed in the Wikipedia thesaurus for achieving this task.

Sentence Position: The system assumes that sentences appearing at the top and bottom of a document have more chances of being important than the rest. Therefore, sentences appearing in the top 20% and the bottom 20% portion of a document are given position scores 50% larger than the others.

3.4 Measuring the Relatedness and Similarity between Sentences

Each sentence would have a vector of the concepts detected in it using the exact match method. When evaluating two sentences, we consider both vectors representing the two sentences to compute the similarity and relatedness between them. The semantic relatedness is computed by the following formula:

$$Srel(Sent1, Sent2) = \frac{rel(A, B)}{PairsCounter}$$

Where Sent1 and Sent2 refer to Sentence1 and Sentence2 respectively, A is the concepts set in Sentence1, B is the concepts set in Sentence2, and PairsCounter is the number of concepts pairs compared.

3.5 Summary Generation

Knowing what features to use in the system, it is possible to assign a score for each feature in each sentence. A sentence score comprises its Topics scores, the relevancy of these Topics with the dominant ones, the overlap between the sentence and the rest of the sentences in a document, and the position of a sentence in the document. After scoring all sentences, the summary is formed by ranking the sentences in a descending order based on their scores, and adding the sentences one by one to the summary till the 100-word limit is reached.

The iterative process can be summarized by the following steps:

1- After scoring all sentences for the first time, we obtain a ranked list of candidate sentences with the top being with the highest score.
2- We remove the top highest scoring sentence from the *Candidate Sentences List (CSL)* and add it to the summary. Only one sentence should exist in the summary at this stage.
3- The cluster of the sentence that was just included in the summary is added to a *Sentences Exclusion List (SEL)*. The cluster should contain the non-simplified version of the sentence in addition to all of its simplified versions.
4- We detect all the *concepts* present in the sentence that was just added to the summary and add them to a *Concepts Exclusion List (CEL)*.
5- We re-score all remaining sentences taking two factors into account: First, sentences in *SEL* should be ignored. Second, any occurrence of a concept that exists in *CEL* should be ignored too.
6- Add the highest scoring sentence to the summary and verify the summary length does not exceed the given limit. If it does not, go to step 3. Otherwise go to the post-processing stage and produce the summary.

Note that in step 5, redundancy is implicitly enforced by counting concepts only once and preferring sentences with a high density of concepts. Simplified sentences that are short and contain important and relevant concepts would still be selected as the approach ensures that no concept repetition within the summary takes place.

After adding the last sentence to the summary and reaching the mentioned word limit, the sentences are re-ordered according to their appearance in the original documents they were taken from. The last sentence in the summary is then truncated to enforce the 100-word limit. At last, we applied a custom set of rules we developed to remove non-important data from some sentences such as date stamps and writers references appearing at the beginning of some sentences.

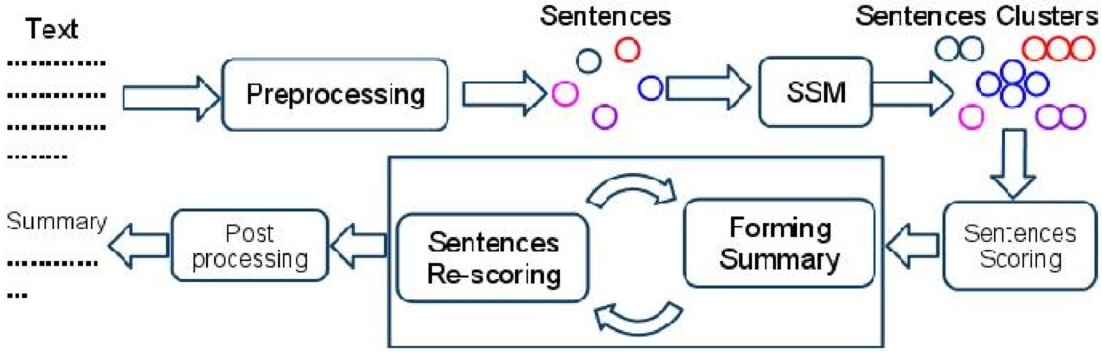

Figure 1: Architecture of the SSM-based Summarizer

4 Evaluations

The provided dataset for the update task is composed of 44 topics divided into five categories. Each topic has a title, category, and 20 relevant documents divided equally into two sets: A and B. Documents in set A precede chronologically those in set B. Participants are asked to submit a summary for each set. They are also given the option of submitting up to two runs for each team.

We participated with two runs. The IDs of our runs are 48 and 44. The SSM-based method was used with run 44 while strong links method was used for 48. In Table 1, the ranks obtained by the system in the different evaluation methods are displayed. The total number of runs the system is compared with is 48.

	Manual	ROUGE-2	ROUGE-SU4	BE
Run 44-A	7	11	26	16
Run 48-A	20	16	29	23
Run 44-B	2	25	8	7
Run-48-B	10	27	11	10

Table 1: Evaluation results for the Update Task showing ranks of the two submitted runs 44 and 48 relative to the 48 submitted runs

5 Conclusion

In this paper, we briefly described the methodology that was implemented in our system for this year's Update task. We outlined how Wikipedia was used, the features that we focused on, and how the summaries were constructed. The results obtained show that the performance of our system is competitive when compared with the other teams systems, although there is still room for improvement. The effect of introducing SSM is also shown in the obtained results.

References:

[1] A. Bawakid and M. Oussalah, "A Semantic Summarization System: University of Birmingham at TAC 2008," in *Proceedings of the First Text Analysis Conference (TAC 2008)*, 2008.

[2] A. Bawakid and M. Oussalah, "Centroid-based Classification Enhanced with Wikipedia," in *The Ninth International Conference on Machine Learning and Applications 2010*, 2010.

[3] L. Qiu, M. Kan, and T. Chua, "A Public Reference Implementation of the RAP Anaphora Resolution Algorithm," *cs/0406031*, Jun. 2004.

[4] A. Bawakid and M. Oussalah, "Using Features Extracted from Wikipedia for the Task of Word Sense Disambiguation," in *9th Conference on Cybernetic Intelligent Systems*, 2010.