Fudan University at DUC 2006

Lin Zhao, Lide Wu, Xuanjing Huang
Dept. of Computer Science and Engineering
Fudan University
{linzhao, ldwu , xjhuang }@fudan.edu.cn

Abstract

This paper describes our work in query-based multi-document summarization task in DUC2006. We present the system overview, focusing on the newly developed techniques, including a new method of sentence similarity calculation, and application of anaphoric resolution to improve the readability of the summary. Evaluation results from NIST are also given and analyzed.

1 Introduction

The DUC task of this year is almost the same as that of last year, which requires creating a brief, well-organized, and fluent summary given a set of relevant documents and a topic in the form of one or more query sentences.

Since the generated summary should be representative of the documents and satisfy the information need described in the given topic, a good method is needed to measure the relevance of the sentences to the topic and how representative the sentences are with respect to the whole document cluster. Therefore sentence similarity becomes an important issue which is widely used as a way of sentence selection in text summarization. Beyond the traditional "bag-of-words" approach which is based on word co-occurrence between sentences, we developed an approach trying to capture the semantic similarities between sentences based on WordNet.

In the following sections, we first give a system overview, and then describe some important steps in detail. Section 3 gives evaluation results from NIST and section 4 concludes.

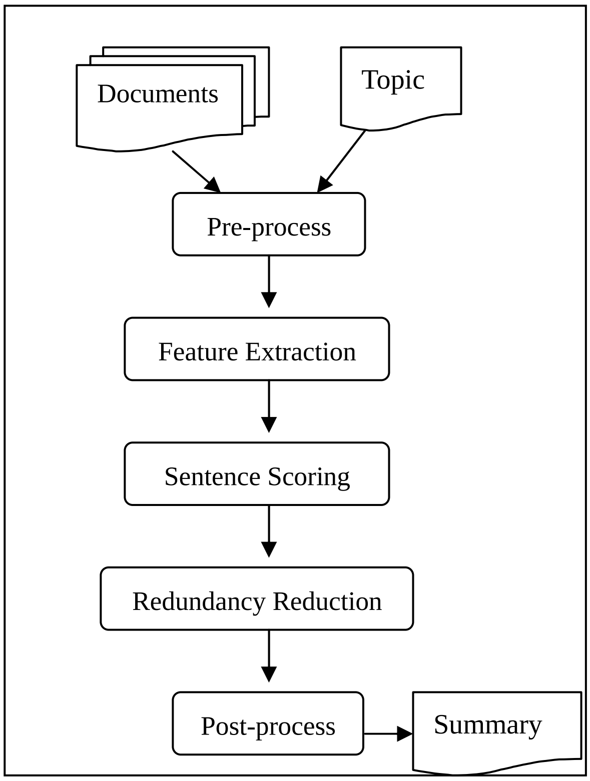


Figure 1. System Overview

The overview of our system is shown in figure 1. Totally there are five steps: (1) pre-processing (2) feature extraction (3) sentence scoring (4) redundancy reduction and (5) post-process.

2 System Overview

2.1 Pre-processing

The pre-processing step includes sentence segmentation, word tokenization and part-of-speech tagging. Besides, we have also done coreference resolution and temporal expression analysis in order to improve the readability of summaries.

2.1.1 Coreference resolution

Existence of pronouns such as *he* or *she* in summaries may leads to referential unclarity without proper contexts. In order to solve this problem, we perform coreference resolution using GATE [1], which can identify reference chains including pronouns. Each chain refers to the same entity. Then for each pronoun on the chains we substitute it with the named entity on the same chain. In this way, pronouns in the original documents can be replaced and not included in the generated summary.

2.1.2 Temporal expression analysis

News articles often include relative temporal expressions, such as yesterday, last year, next month etc. Once such expressions are included in the summary, they become meaningless. So we developed a module to deal with this problem. First, relative temporal expressions are identified from the text, and the publication date of this article is also obtained from the original text information. Then we use predefined rules to replace such relative dates with their absolute values. When doing replacement, prepositions are sometimes required to add in order to maintain the correct grammar of the sentence. For example, "… happened last year" is rewritten as "… happened in 1999" assuming the publication date of the article is 2000.

2.2 Feature Extraction

In this step, we extract features from each sentence and use them to evaluate the sentence. Two kinds of features are used here. One is sentence similarity between the candidate sentence and the rest of document cluster, which is used as measurement of how representative the sentence is of the whole cluster. The other is sentence similarity between the candidate sentence and the given topic, which is used as measurement of how relevant the sentence is to the topic. Based on combination of above two features, we hope the extracted sentences can meet the users' information need. Since sentence similarity is a key point here, we try to develop an effective approach to solve this problem. We observed that using the traditional word-based approach makes many sentences have zero similarities with other ones, especially with the given topics because of no word co-occurrence between them, although they are semantically related. Therefore we developed an approach for sentence semantic similarity calculation based on WordNet.

2.2.1 WordNet relations

WordNet [2] provides relations between words such as synonym (eg, car ~ automobile), hypernym (eg, car ~ vehicle), holonym (eg, accelerator ~ car), etc. These relations are useful for us to connect different words together if they are semantically related. We do not use all the relations in WordNet, however, only part of them are selected based on the experimental results.

For noun, we select five relations including synonym, hypernym, holonym, derivation (eg, crime ~ criminate) and domain (eg, crime ~ smuggle). The use of derivation relation is to find for a word its derivationally related forms. For verb, we select four relations including synonym, hypernym, derivation and domain. For adjective and adverb, only synonym is used. These relations serve as basis of our sentence representation.

2.2.2 Sentence representation

Sentence representation is the basis for sentence similarity calculation. Given a sentence, a good representation should be able to capture its implicit semantic information. Due to the lack of a good semantic parser, we use WordNet as our semantic resource to build a semantic-based vector instead of word-based one for a sentence. That is, for each content word (noun, verb, adjective and adverb) in a sentence, we try to find its correlative concepts in WordNet based on the above relations and then add them into the original vector. The main idea is to extend the initial word-based vector by adding relevant concepts into it. After such extension, two sentences which are originally unrelated may be related together. The detailed procedure is as follows.

1. An initial sentence vector *(w_1, w_2, …, w_n)* is built which consists of content words in the sentence.
2. Search in WordNet for each word w_i in the vector.
3. If w_i does not exist in WordNet, keep it in the vector and deal with the next word. The reason to keep this word is that such words that do not exist in WordNet are usually named entities and can carry important information, so they should not be removed from the vector.
4. If w_i exists in WordNet, add into the vector the concept entries *(c_{i1}, c_{i2}, …, c_{ik})* which are extracted from WordNet using the relations described in section 2.2.1. That is, for the word w_i, we obtain its synonym, hypernym, holonym etc if they exist and add them into the vector. Since a word may have more than one sense, we have tried two methods for doing word sense disambiguation (WSD). One is to use "first sense" method in which the most frequent sense of each word in WordNet is selected. The other is as the algorithm described in [3], which determines word senses in the given context. However, experimental results on test data show that the latter does not gain significant improvement over the former, which is probably because the performance of WSD is not good enough to satisfy the requirement. Considering the system efficiency, we choose to use "first sense" method in official run. Furthermore, when doing vector extension using the WordNet relations, we only consider one level of the relations. Higher levels are not considered in order not to introduce noise into the vector since the higher the level, the less related the words are.
5. Get final vector representation *(w_1, w_2, …, w_n, c_{11}, c_{12}, …, c_{1k}, ….., c_{n1}, c_{n2}, …, c_{nm})* for the sentence.

In this way, we can obtain semantic information that is not explicitly expressed in the sentences. Therefore, sentences with different but semantically related words can be related together.

2.2.3 Sentence features

As we have mentioned before, two kinds of features are extracted from each sentence. Given a sentence S_i, one feature is its average similarity with the rest sentences. It is calculated as follows.

$$F_1(S_i) = \frac{1}{N-1} \sum_{S_j \in D - S_i} sim(S_i, S_j)$$

where $sim(S_i, S_j) = \cos(\vec{S}_i, \vec{S}_j)$, $\vec{S}_i$ represents the sentence vector derived from previous steps. D is the set of sentences in the document cluster, and N is the total number of sentences.

The other feature is the similarity between the sentence S_i and the topic. Since a topic may consist of multiple queries, we compute similarity of S_i with each query, and choose the maximum result, as follows.

$$F_2(S_i) = \max_{q_j \in Q} sim(S_i, q_j)$$

where Q is a topic and q_j is the j-th query in the topic. $sim(S_i, q_j)$ is also obtained using cosine calculation except that in vector representation for the queries, only synonym relations are used, because our experimental results show that adding other relations leads to worse performance. A possible explanation is that queries are often composed of abstract words, so it is not necessary to add more abstract words such as its hypernym into the vector, which may contribute less or even negatively during similarity calculation.

2.3 Sentence Scoring

The score of a sentence is used to measure how important a sentence is to be included in the summary and it is calculated as the weighted linear combination of the above two features.

$$Score(S_i) = wF_1(S_i) + (1-w)F_2(S_i)$$

where w is the feature weight and in our system it is empirically set to 0.8.

2.4 Redundancy Reduction

After we get sentence scores, the module of redundancy reduction is carried out to extract sentences and generate a summary. This module is almost the same as the one in our system of last year [4].

In each iteration, we extract the sentence with the highest score, and then adjust scores of the remaining sentences. Scores of sentences that are very similar with the extracted sentence are adjusted downwards in this way. This process is repeated until we reach the length restriction of the summary.

2.5 Post-process

After a summary is generated, sentence re-ordering is performed to ensure the coherence of the summary. We applied a simple method which groups similar sentences together based on intra-sentence similarities calculated in previous stage because we think similar sentences are often topic-related and should be put together.

3 Evaluation

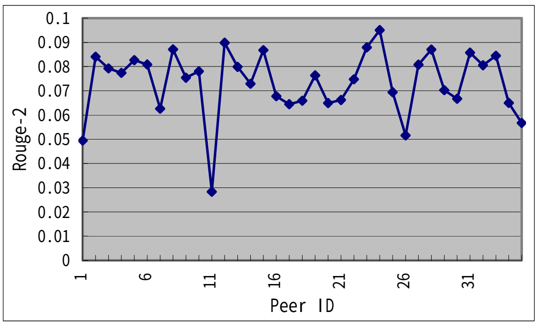
3.1 Data and Evaluation Metrics

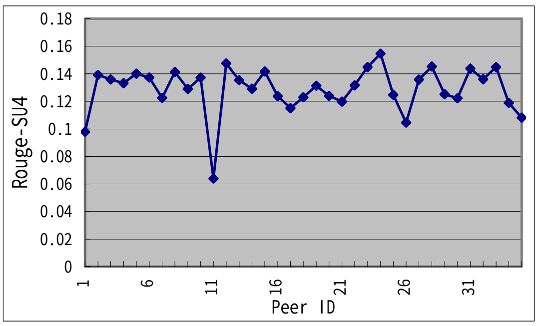
50 document clusters from AQUAINT corpus are used as evaluation data and each cluster contains 25 documents. For each cluster, 4 manual summaries are provided for evaluation.

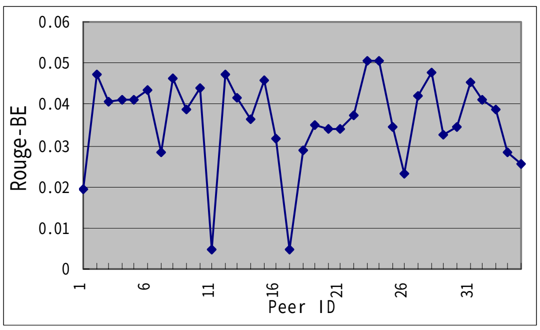
The peer summaries are evaluated both automatically and manually, including automatic Rouge evaluation [5] and manual evaluation on responsiveness and linguistic quality of the summary.

3.2 Evaluation results

Evaluation results of all systems are shown in Figure 2. Among the 34 participants, our system ranks 11th in both Rouge-2 and Rouge-SU4 evaluation, the 10th in BE evaluation, the 9th and 11th in overall responsiveness and content responsiveness respectively, and the 7th in pyramid evaluation out of the 22 systems. While our system performance on linguistic quality is rather poor which only ranks 27th. We analyzed results on the five quality questions and found that our performance on Q3 (referential clarity) is improved compared with that of last year. This is most likely due to the reference rewriting technique that we have used in the pre-process step. However, the poor performance on other questions is needed for further exploring.







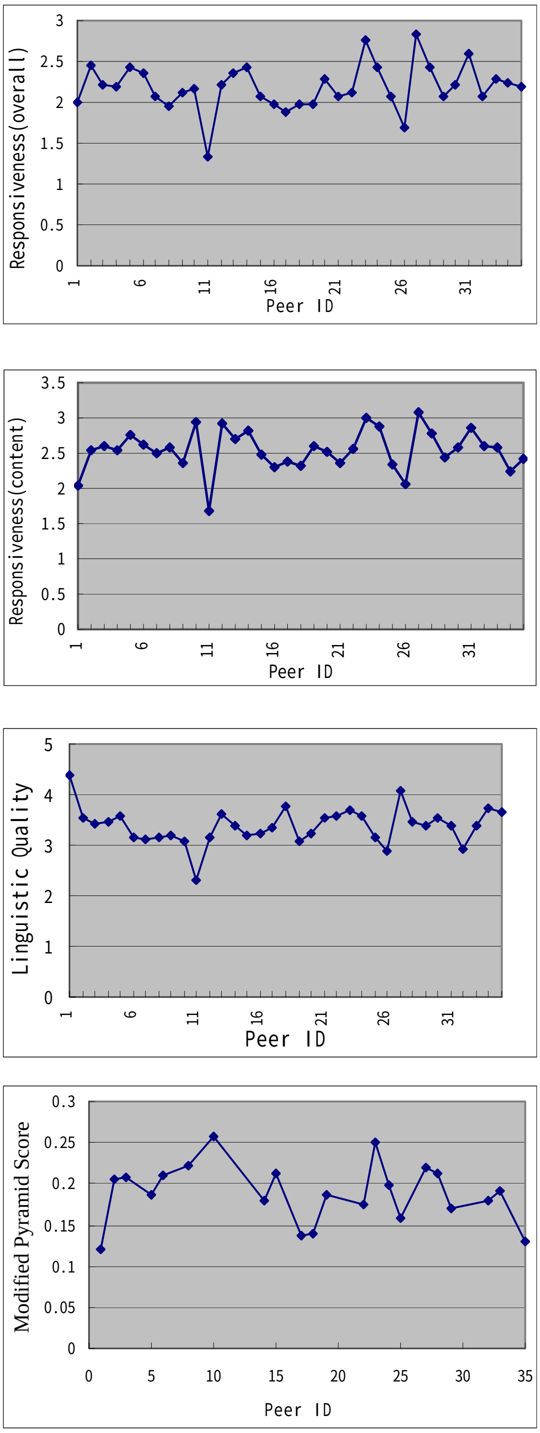


Figure 2. Evaluation results of all systems

4 Conclusion and Future Work

In this paper, we described our participation in DUC 2006. Through experimentation, we find that there're still lots of rooms for improvement. A wide-coverage coreference tool will be useful when doing coreference resolution. An effective word sense disambiguation method may also benefit the system when calculating sentence similarity. Moreover, we'd like to focus more on how to improve the linguistic quality of summaries.

5 Acknowledgments

This research was partly supported by NSF of China under contract of 60435020. We are very thankful to Xian Qian, Zhongqin Wu, Ningyu Chen for their contributions in our work.

References

[1] GATE package, http://gate.ac.uk/

[2] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five Papers on WordNet. Technical report, Princeton University, 1993.

[3] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In Proc. of the SIGIR 2003 Semantic Web Workshop, 2003

[4] L. Zhao, X. Huang, L. Wu. Fudan University at DUC 2005. In Proceedings of Document Understanding Conference (DUC 2005), Vancouver, BC Canada.

[5] Lin Chin-Yew and Franz Josef Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In Proceedings of ACL 2004