PolyU at TAC 2008

Wenjie Li, You Ouyang, Yi Hu, Furu Wei

Department of computing, The Hong Kong Polytechnic University
{cswjli, csyouyang, csyhu, csfrwei}@comp.polyu.edu.hk

This paper presents the participation of the Hong Kong Polytechnic University in the TAC 2008 competition. The systems for the participated tracks are introduced respectively.

1 Introduction

PolyU has participated in three of the TAC 2008 tasks, including the update text summarization track, the opinion text summarization track and the query answering track. We submitted three independent systems for these tracks. Thus the details of the opinion summarization track and the QA track are introduced in the following sections respectively.

2 Opinion Summarization Track

2.1 System Overview

The opinion summarization track is a complex task which involves query answering, multi-document summarization and opinion analysis. It requires producing short coherent summaries of the answers to some opinion-oriented questions from associated blog documents. Text snippets of the documents output by QA systems are also provided. We follow a typical feature-based framework to build systems to adapt the multi-purpose task. The system is implemented by extracting the salient sentences from the original documents to form the summary. The sentences are selected through a two-phase process considering both the paragraphs and sentences.

We submitted two runs to the track, one using the provided snippets and one not. Both systems consist of three modules, i.e. the candidate sentence retrieval module which transforms the original input data to candidate sentences, the sentence scoring module which identifies the salient sentences form the candidates, the summary generation module which generates the summary from the select sentences. Figure 1 illustrated the structure of the systems.

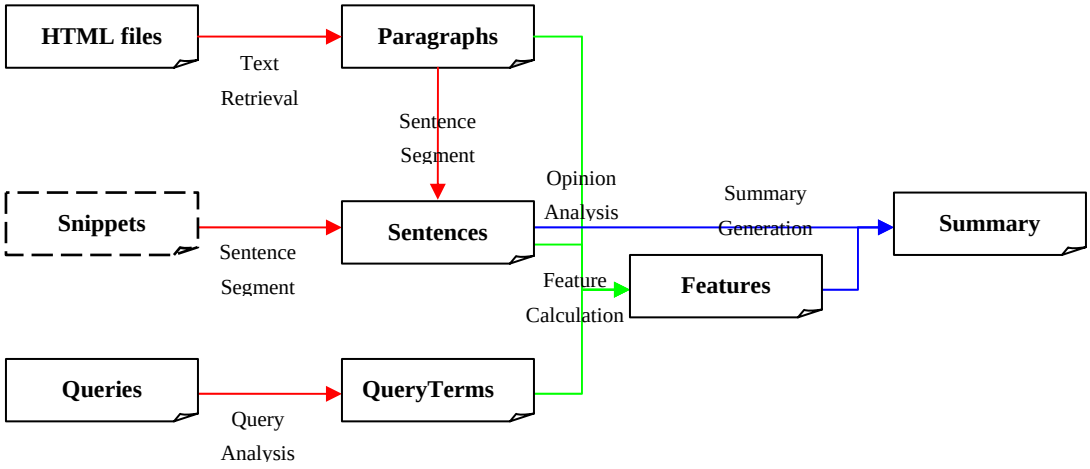

Figure 1. System structure

2.2 Candidate Sentence Retrieval

The objective of this module is to retrieve the candidate sentences from the HTML files for later processes.

2.2.1 Retrieving without snippets

The original documents in the TAC 2008 data set are in HTML style. Therefore, the very first step of the systems is to retrieve the texts from the HTML files. In our system, we rely on several heuristics rules based on HTML tag analysis to identify the text segments, detailed as below:

(1) Only the section appearing between "<body>" and "</body>" is considered;

(2) Each section of text between " " and " " or ">" and "
" are retrieved and regarded as a paragraph;

(3) The first html tag contains "comment" such as "<begin-comment>", "<comments>" or etc is regarded as the boundary of the original post from the blogger and the comment posts from the visitors.

2.2.2 Retrieving with snippets

The above rules are very simple thus it can not retrieve all the texts in the documents. The snippets can be used to make complement by two ways:

(1) The text pieces in the HTML files which contain the snippet segments;

(2) The snippets themselves.

2.2.3 Sentence Segment

All the retrieved texts are fed to the GATE tool [4] to be segmented to sentences. The output sentences of GATE are used to compose a sentence candidate collection through a simple filter which removes the sentences with less than 6 words or very low grammatical quality.

2.3 Sentence Scoring

After the sentence candidates are obtained, the next step is to determine the salient sentences to be included in the summary.

2.3.1 Scoring without snippets

In our systems, we regard the informative richness as the main criteria for measuring the importance of a sentence, considering how much useful information it contains for the queries. Since the task involves opinion analysis, QA and document summarization at the same time, various characteristics of the sentences should be considered to measure their importance. In our system we use multiple features to directly depict the characteristics, including **Centroid**, **SimtoQuery**, **PosSentiment**, **PosOpinion**, **NegSentiment**, **NegOpinion** and **Position**. Centroid and SimtoQuery are two traditional features in the summarization area which reflect the informative richness of the sentence to the text collection and the query respectively [5]. For this opinion-oriented task, four features are also included to depict the informative richness of the sentences in expressing opinions. The features are simply defined by the numbers of the words appearing in several dictionaries, including the positive or negative dictionaries of sentiment terms and opinion terms. At last, a position feature is simply used to indicate if the sentence belongs to the original post field or the comment field since we thought that the comments may be more likely to express opinions and views than the original post.

In the blog posts or comments, spoken language usages are very common since blogs are not as formal as newswire or literature articles. The sentences in the blog posts or comments may possibly be not complete. Therefore, a single sentence may be not sufficient for describing an opinion or a view completely. Motivated by this, we introduce several paragraph-level features to complement the sentence-level features, including ParaCentroid, ParaSimtoQuery, ParaOpinion, ParaSentiment. The definition of these features is very similar to the sentence-level features. The difference is that they are defined on the paragraphs.

Using the above features, a composite score is defined as a linear combination of the features which aims at synthesizing the effect from different aspects of the sentence.

2.3.2 Scoring with snippets

To the system using the snippets, an additional SimtoSnippet feature defined as the similarity between the snippets and the sentence is also included. Since the snippets contain the most important information of the texts, this feature is regarded as a decent estimation of the sentence importance.

2.4 Summary Generation

1.4.1 Sentence selection strategy without snippets

Instead of directly using the composite score to select the sentences, we adopt a two-phase strategy to select the desired sentences. The selection process starts with select candidate paragraphs using the paragraph-level features. In the above retrieving process, a retrieved paragraph is either a paragraph from the blogger poster or a paragraph from one commenter. Therefore, we believe a paragraph can be regarded a complete semantic unit which reflects an opinion from one person. Therefore, we only consider the sentences appeared in the informative paragraphs as potential sentences to be included in the summary. The informative degree of a paragraph is reflected by its ability to answer the queries, such as similarity to query, opinion significance and etc. So in this step, we remove all the uninformative paragraphs which do not include enough query terms or opinion terms. Then the remained sentences are ranked by the composite scores. The composite scores of all the sentences are normalized by the maximum one. The sentences whose scores are more than 0.3 are selected into the summary and a maximum of 20 sentences is allowed.

2.4.2 Sentence selection strategy with snippets

In the system with snippets, some additional sentences are also selected from the retrieved sentences with snippets. First, the sentences which are similar to the snippets are selected. The similarity is calculated by the overlapping rate of the snippets and the sentence and the threshold of the overlap rate is set to 0.6. At most 15 sentences of this type can be added in the summary. At last, the sentences directly appearing in the snippets are also added to the summary. The maximum number for this type of sentences is also 15.

2.4.3 Post Processing

In summarization from multiple resources, the original texts usually contain redundant information. Thus we adopt the Maximum Marginal Relevance approach to remove the repetitive sentences. The sentences are selected iteratively that each

round the candidate sentence will be selected only when it is not too similar to any sentences already in the summary. At last, all the selected sentences are refined to increase the readability of the summary. The refinement includes several simple rules for removing irregular tokens and capitalizing the headword.

2.5 Evaluations and Results

In the opinion summarization track of TAC 2008, a total of 36 runs are submitted. Among these submission, 17 systems used the given snippets and 19 systems not. Because the snippet files provided additional information, generally the system using the snippets performed better in answering the questions. We submitted two runs to the track, i.e. the Polyu1 system which did not use the snippets and the Polyu2 system which used the snippets. The results of the TAC evaluations are listed in Table 1. The BestAll and BestNoSni indicate the best result in all the submitted systems and in all the systems without the snippets respectively. Table 2 shows the ranks of our systems in all the systems. In our belief the two kinds of systems using or not using snippets should be compared respectively. Therefore, we provide the ranking results both in all the systems and each kind of systems. In the table, the first number indicates the rank of the system in the corresponding kind of systems and the second one stands for the rank in all the systems.

Table 1. Results of the evaluations

System	Pyramid F-score	Gramma-ticality	Non-redundancy	Structure Coherence	Overall readability	Overall responsiveness
Polyu2	0.380	5.636	5.727	3.091	4.136	4.545
BestAll	0.534	7.545	8.045	3.591	5.318	5.773
Polyu1	0.251	5.864	6.909	3.045	3.864	2.864
BestNoSni	0.251	6.000	7.909	3.591	5.318	3.909

Table 2. Ranks of the two systems

System	Pyramid F-score	Gramma-ticality	Non-redundancy	Structure Coherence	Overall readability	Overall responsiveness
Polyu1	1, 11	3,7	2,4	6,15	4,13	6,19
Polyu2	6, 6	4,9	10,23	6,11	4,6	7,7

In the results, the Polyu2 system performed better than Polyu1 since generally the systems using the additional snippets performed better. However, the Polyu1 system performed relatively better than Polyu2 in corresponding kind of systems. This is

because we mainly focused on retrieving the salient sentences from the original HTML files. The usage of the snippets was very simple in the Polyu2 system. The most different performance of the two systems is in the Non-redundancy evaluation. This is because the Polyu2 system selected more sentences from three different resources thus it suffered more risk of containing repeating information.

3 Question Answering Track

3.1 Introduction

The goal of the QA task is described as developing a system that retrieves precise answers to questions by searching large document collections from the corpus Blog06. Different from the QA tracks in previous TREC, the QA task in TAC08 requires participating systems providing answers to the questions with opinion.

The opinion questions have two types, each with its own evaluation metric. The first one is rigid question, such as “Name US senators who support tax reform”, which needs strings containing a list term as the answers of such questions. The evaluation metric is F-measure combining precision and recall. The second one is squishy question, such as “Why do countries want to have nuclear power plants?”, which needs strings to be answer string to the question. Its evaluation metric is nugget pyramid evaluation used for "Other" questions in the TREC 2006-2007 QA track.

In terms of our approach in TAC 2008, there are four sequential steps in extracting answers. Briefly to say, the first one is a preprocessing procedure that contains tagging the opinion of a question. This is a human intervention procedure. We tagged the opinion of each question with “Polarity = Positive” or “Polarity = Negative”. This preprocessing step also extracts text contents from the document collections, and then split the text contents into sentences. The answers are assumed to hide in these content sentences. In this approach, we only use the provided top 50 blog pages to each question as document collection. The second step recognizes the polarity of a sentence in these text contents. We believe that the answers should be in the sentences with the same sentiment orientation of the opinion of the correlated question. The third step calculates the similarity between a sentence and the correlated question. This process filters out the sentences with less possibility to answer the question. The last step extracts the answers from the top ranked sentences. The sentences with higher rank are assumed to more possibly contain the answers.

The following sections will introduce the details used in these steps respectively.

3.2 Preprocessing Procedures

3.2.1 Tagging Questions

The questions of QA track in TAC 2008 have their sentiment orientation. For example, the rigid question "Who prefers Starbucks to Dunkin Donuts?" has positive polarity and the squishy question "What features do people dislike about Vista?" has negative polarity. In this step, we give a sentiment label to all questions. The polarity label will be read into memory as an attribute of its correlated question. These labels are used to guide the next steps to collect sentences with the same orientation from the top 50 documents.

3.2.2 Extracting and Segmenting Text Contents

Because the data of Blog06 come from Blogs, the web pages are organized with hierarchical structure by using defined XML tags.

In our system, we only extract the contents between the tags and as the text contents. Seen from the extracted contents, their also exit many non-letter characters in them. After getting the text contents, we use the character ".", "!", "?" and ";" as the end mark of sentences to split the text contents into sentence sequences. The answers are regarded to be hidden in these individual sentences.

3.3 Language Modeling Approach to Sentiment Classification of Sentences

To handle the problem of recognizing the sentiment orientation of a sentence, we propose an idea to estimate both the positive and negative language models from training collections. Then the orientation of a sentence is computed by the Kullback-Leibler divergence between the language model estimated from the sentence and these two trained sentiment models. We assert the polarity of a sentence by observing whether its language model is close to the trained "Positive" model or the "Negative" model.

The motivation of language modeling is simple: the "Positive" and "Negative" languages are likely to be substantially different, i.e. they prefer to different language habits. We exploit this divergence in the language models to classify a sentence.

The "Positive" orientation is represented with a positive language model θ_P that is a probability distribution over N-grams in positive training collection. Accordingly, a negative language model θ_N represents the language model for "negative" orientation. A test sentence generates a language model θ_s. Note that a language model is a statistical model: probability distribution over language units, indicating the

likelihood of observing these units in a language. Therefore a sentence can then compare its model with "Positive" or "Negative" model using distance mechanism:

$$\varphi(s;\theta_P,\theta_N)=Dis(\theta_s,\theta_P)-Dis(\theta_s,\theta_N):\begin{cases}<0 & Positive\\ >0 & Negative\\ =0 & Neutral\end{cases}. \qquad (1)$$

Where *Dis(p,q)* is the distance between two distributions *p* and *q*. This formula expresses the classifying idea that if $Dis(\theta_s,\theta_P)$ is smaller than $Dis(\theta_s,\theta_N)$, it means the test sentence *s* is closer to "Positive". Otherwise, if $Dis(\theta_s,\theta_P)$ is greater than $Dis(\theta_s,\theta_N)$, "Negative". Note that if $\varphi(s;\theta_P,\theta_N)$ equals to zero, the test document is regarded to be "neutral". Then, we exploit the Kullback-Leibler Divergence as the distance measure.

$$\begin{cases}Dis(\theta_s,\theta_P)=\sum_{ngram\in s}\Pr(ngram\mid\theta_s)\log\left(\frac{\Pr(ngram\mid\theta_s)}{\Pr(ngram\mid\theta_P)}\right)\\ Dis(\theta_s,\theta_N)=\sum_{ngram\in s}\Pr(ngram\mid\theta_s)\log\left(\frac{\Pr(ngram\mid\theta_s)}{\Pr(ngram\mid\theta_N)}\right)\end{cases} \qquad (2)$$

All the details please refer to [1].

3.4 Exploring Similarity between Question and Sentence

This section discusses the similarity between a question and a sentence, and this similarity is explored to filter out the sentences which have less possibility to contain an answer. The assumption is that an answer containing sentence ought to be similar to the correlated question.

Somers [3] reviewed several sentence distance or similarity measures that were linguistically motivated. Different linguistic components of a sentence (e.g. characters, words, or structures) can be used as comparison units. So far, character-based matching, word-based matching, structure-based matching, and syntax-matching have been used.

In our system, we treat a sentence as a pure string [2]. First, since the string view allows the system manipulation of many languages as pure strings, we can use the same algorithm to retrieve sentences written in different languages. Second, each sentence can be treated as a small piece of document and sentence similarities can be adapted to rank sentences. Third, in a language-learning context, sentence

correctness is difficult to predict and pure string comparison is a more tolerant approach than a linguistic component-based approach.

This system explorers the Dice Coefficient method, and the Dice Coefficient here is an N-gram-based (exactly to say, unigram and bigram) similarity measure. The similarity value is related to a ratio of the number of common N-grams for both two strings and the individual number of total N-grams of the two sentences respectively. When comparing a question *Q* and a sentence *S*, if n_{com_uni} (n_{com_bi}) is the count of common unigrams (bigrams), n_{Q_uni} (n_{Q_bi}) is the total count of unigrams (bigrams) of question *Q*, and n_{S_uni} (n_{S_bi}) is the total count of unigrams (bigrams) of sentence *S,* the Dice coefficient can be expressed as follows.

$$Dice(Q,S) = \alpha_1 \frac{n_{com\_uni}}{n_{Q\_uni} + n_{S\_uni}} + \alpha_2 \frac{n_{com\_bi}}{n_{Q\_bi} + n_{S\_bi}} \qquad (3)$$

where $\alpha_1 + \alpha_2 = 1$.

In the similarity measure, question and sentence are decomposed into smaller unigram and bigram units. All grams are used as elements representing *Q* and *S*. Obviously, the different values of the two alphas will balance the contributions of the unigrams and bigrams. The system simply sets $\alpha_1 = \alpha_2 = 0.5$.

3.5 Answer Extraction

In sentence ranking, we extract from the entire collection of web pages <*Q*, *S*, *P*, *Sim*> triples. They are respectively a question *Q*, a sentence *S* of the question, their common sentiment orientation *P*, and the similarity representing its likelihood of containing an answer.

In terms of a *Q*, we rank its sentences appearing in its correlated triples by using the corresponding similarities. That means we filter out unlikely sentence candidates.

For the answer extraction of rigid question, the first noun phrase with capital letter will be extracted from the top sentences to organize the answer set. If some sentence does not contain such noun phrase with capital letter, the system will go on extracting the answer from the next one. At last, the system extracts at last 5 answers for each rigid question. As an alternative, the system also tries to extract the first noun phrase which not only begins with capital letter but also is a named entity. The recognition of named entity is implemented by using the famous GATE tool. This answer set with named entity recognition is submitted as the PolyU2 result, that is, the second run.

For the answer extraction of squishy question, the system output the first 10 sentences as the answers. This is a simple method to get the answers to a squishy question.

4 Conclusion

In this paper we described our systems for the TAC 2008 tracks. The proposed two-phase opinion summarization system performed well in the opinion summarization track.

References

[1] Hu, Y., et al.(2007) A Language Modeling Approach to Sentiment Analysis. In Proceedings of ICCS 2007, Lecture Notes in Computer Science, Vol. 4488, pp.1186-1193, Springer-Verlag.

[2] Jin, Z., Barrière, C. (2005) Exploring Sentence Variations with Bilingual Corpora. Corpus Linguistics 2005 Conference. Birmingham, United Kingdom. July 14-17, 2005.

[3] Somers, H. (1999) Review Article: Example-based Machine Translation, Machine Translation, vol. 14, 113-157.

[4] http://gate.ac.uk/

[5] Multi-document Summarization Using Support Vector Regression. Sujian Li, You Ouyang, Wei Wang, Bei Sun. In Document Understanding Conference 2007.