BJUT at TREC 2014 Temporal Summarization Track

Yun Zhao, Fei Yao, Huayang Sun, Zhen Yang*
College of Computer Science, Beijing University of Technology, China
yangzhen@bjut.edu.cn

Abstract

This paper describes the second participation of BJUT in the temporal summarization track. We performed the experiments on the TREC KBA 2014 stream corpus using the classic information retrieval models, such as BM25, vector space model. Also, we introduce the details of our system, which consists of corpus pre-processing, information retrieval module and information process module.

Introduction

The TREC Temporal Summarization Track runs for the second time in this year, and different from 2013[1], this year's track focuses on only one task: Sequential Updates Summarization. All participants should answer a query based on topic a set of relevant and novel sentences ranked by time from a time-ordered stream of documents, through which users can efficiently monitor the information associated with an event (such as a natural disaster) over time[2]. Another primary difference lies in the data size, which reduced to 559G from 4.5T. The corpus, namely TREC-TS-2014F, is a filtered version of the full track. It is designed for using by participants of the TREC-TS track, aiming to provide a dataset with which groups can participate in the TREC-TS track without having to process the full corpus. The corpus consists of a set of time stamped documents from a variety of news and social media sources covering the time period from Oct.2011 to Apr. 2013. A document contains a set of sentences, each with a unique identifier.

SYSTEM

According to the task of Temporal Summarization Track, system should emit relevant, important and novel sentences to a specific event. We submitted three runs for this task and the implementation framework of our system is shown in FIG. 1.

As shown in FIG. 1, the framework of our temporal summarization system can be described as follows, which mainly includes corpus pre-processing, information retrieval module and information process module.

- Information pre-processing module
  The corpus downloaded locally from $streamcorpus-2014-v0\_3\_0-ts-filtered$[3] is encrypted file, which cannot be used directly. In this sense, firstly, decrypting the files uses the authorized key and converts the .GPG file format to .SC file format; Secondly, parsing the .SC files use stream corpus toolbox to .TXT files. The stream corpus toolbox is given by TREC and provides a common data interchange format for document processing pipelines that apply language-processing tools to large streams of text.
- Information retrieval module
  Firstly, building index for the .TXT files. Then, combining with query expansion for retrieval to get relevant sentences.
- Information process module
  Text similarity and clustering can improve the accuracy and recall rate of the retrieval results. We used two methods to complete this part. After the topic clustering, the centers of the different clustering are chosen to build the summarizations. Then the summarizations are ranked by time factor and similarity factor.

The frame with solid line in FIG. 1 is the main methods we used for the temporal summarization track. The details of our work will be introduced in the next section. We emphasis on the description of the two key parts: information retrieval module and information process module.

Information Retrieval Module

- Information Retrieval
  In this part, we use Lemur[4] for information indexing and retrieval. Lemur is a toolkit designed to facilitate research in language modeling and information retrieval (IR). It supports the construction of basic text retrieval systems using language modeling methods such as BM25 [5]. Our experiment has two steps to build the index. First, create a parameter file tell the lemur toolkit how to index; Secondly, use IndriBuildIndex.exe application to build index. Accordingly, the realization of retrieval also has two steps. First, create a parameter file tell the lemur toolkit how to retrieve; Second, use IndriRunquery.exe application to retrieve.
- Query expansion
  Generally, when we retrieving information based on query words, there always exists one problem, namely word mismatch, which can be explained that people often use

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2014		**00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
BJUT at TREC 2014 Temporal Summarization Track	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Beijing University of Technology,College of Computer Science,Beijing 100124, China,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
presented in the proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014) held in Gaithersburg, Maryland, November 19-21, 2014. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA).

14. ABSTRACT
This paper describes the second participation of BJUT in the temporal summarization track. We performed the experiments on the TREC KBA 2014 stream corpus using the classic information retrieval models, such as BM25 vector space model. Also, we introduce the details of our system, which consists of corpus pre-processing information retrieval module and information process module.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**4**	

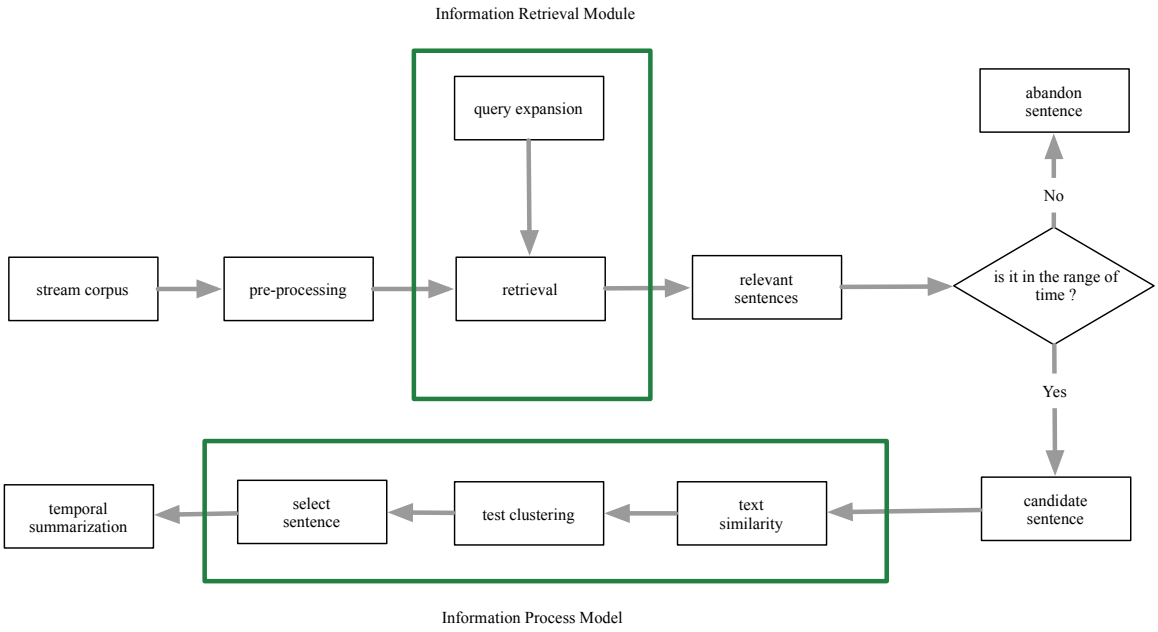

Figure 1: The framework of the temporal summarization track.

different words to describe the same concepts between the queries and documents. To solve this problem, in this paper we used a method called query expansion. Query expansion can augment a query word, and through which, the query word can match more sentences, thus potentially increasing the number of relevant results. The query word is extended by using words with similar meaning to those in the query, and the chance of matching words in the relevant documents is therefore increased.

Information Process Module

After information retrieval module, we got a set of sentences related to a topic. Considering the large amount of these sentences, in order to simplify the computation, we first judge whether these sentences are in the range of each topic's begin time and end time. If a sentence is not in the time period, abandon it, and the rest sentences can be treated as candidate sentences, which should be had further processes such as text similarity calculation and text clustering.

- Text Similarity
  We used two methods based on Vector Space Model (VSM)[6], which is composed of eigen values extracted from documents and its weight. Vector Space Models is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings. In VSM, sentences and queries are represented as vectors:

$$d_i = (w_{1,j}, w_{2,j}, \cdots, w_{t,j}) \tag{1}$$

Each dimension corresponds to a separate term. If a term occurs in the sentence, its value in the vector is non-zero. There are several ways to compute these values (also called weight), and in this paper, we use tf-idf weighting. The VSM model is known as the term frequency-inverse document frequency model. The weight vector for document distance:

$$V_d = (w_{1,d}, 2_{2,d}, \cdots, w_{N,d}) \tag{2}$$

Where

$$w_{t,d} = tf_{t,d} log(\frac{|D|}{|\{d' \in D | t \in d'\}|}) \tag{3}$$

And $tf_{t,d}$ is term frequency of term t in document d (a local parameter), $log \frac{|D|}{|\{d' \in D | t \in d'\}|}$ is inverse document frequency (a global parameter). $|D|$ is the total number of documents in the document set; $|\{d' \in D | t \in d'\}|$ is the number of documents containing the term t.

After getting the vectors, the VSM similarity between two documents can be calculated by using the cosine distance:

$$sim(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i|| \cdot ||d_j||} \tag{4}$$

Another method we used to calculate similarity is based on mutual information preserving mapping (MIPF), which is a manifold learning algorithm that computes low-dimensional, neighborhood-preserving based on mutual information of high-dimensional inputs. With sufficient data set, we expect each document text can be expressed as its neighbors' mutual information and its neighbors are lie on or close to a locally linear patch of the manifold. Then each text data can be reconstructed from its neighbors which are based on information content. Reconstruction errors are measured by the cost function

$$\Sigma_i |I(X_i) - \sum_j W_{ij} I(X_i; X_j)|^2 \tag{5}$$

Table 1: Experimental Result.

		EG			C			F		
		Q0	Q1	AVG	Q0	Q1	AVG	Q0	Q1	AVG
Topic	11	0.0504	0.0396	0.0358	0.1030	0.0962	0.3221	0.0677	0.0561	0.0552
	12	0.0171	0.0176	0.0096	0.2367	0.2341	0.1986	0.0320	0.0327	0.0168
	13	0.0538	0.0570	0.0172	0.6300	0.6295	0.4463	0.0992	0.1046	0.0314
	14	0.0239	0.0271	0.0051	0.1833	0.1963	0.3317	0.0423	0.0477	0.0094
	15	0.0701	0.0732	0.0439	0.7267	0.6977	0.7259	0.1278	0.0477	0.0094
	16	0.0895	0.1253	0.0634	0.9812	1.0405	0.8336	0.1278	0.1326	0.0754
	17	0.0414	0.0482	0.0240	0.6024	0.5279	0.6562	0.1641	0.2237	0.1146
	18	0.0445	0.0358	0.0210	0.2479	0.1989	0.4575	0.0754	0.0607	0.0388
	19	0.0796	0.0938	0.1043	0.4695	0.5284	0.4601	0.1361	0.1593	0.1237
	20	0.0535	0.5552	0.0172	1.0214	1.0214	0.8989	0.1016	0.1047	0.0332
	21	0.0955	0.0626	0.0341	0.4137	0.3841	0.4648	0.1551	0.1077	0.0552
	22	0.1046	0.0973	0.0773	0.5108	0.5009	0.4625	0.1737	0.1629	0.1267
	23	0.0805	0.0805	0.0612	0.1235	0.1235	0.1911	0.0975	0.0975	0.0840
	24	0.0739	0.0739	0.0373	0.3847	0.3847	0.3589	0.1240	0.1240	0.0633
	25	0.0992	0.0992	0.0329	0.4483	0.4483	0.4516	0.1624	0.1624	0.0570
Mean	ALL		0.0389			0.4840			0.0620	
	Q0		0.0652			0.4722			0.1091	
	Q1		0.0658			0.4675			0.1110	

In order to minimize the reconstruction errors, we can get the weight W. By using the weight, the low-dimensional vector Y can be measured by the embedding cost function:

$$\sum_i |I(Y_i) - \sum_j W_{ij} I(Y_i; Y_j)|^2 \quad (6)$$

Unlike traditional manifold methods for images, MIPF applies on text field, and its optimizations use the relationship between different texts. Meanwhile it remains the advantages of the manifold method. By exploiting the local symmetries of reconstructions, MIPF is able to learn the global structure of mutual-information-based manifolds, such as those generated by documents of text.

- Text Clustering
  The k-means [7] clustering is chosen after many experiments. The k-means clustering is a popular method for cluster analysis in data mining. The k-means clustering aims to partition n observations into k clusters, in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

- Sentence Selection
  After text clustering, we can get the clusters based on topics between different events for information expansion. We choose the centers of the clusters and the top sentences as the summarization. Finally each event we totally choose about 100 sentences from the thousands sentences. The last step is to rank these central sentences. Time and similarity are the two factors that used to rank the summarizations. After this step, the final temporal summarization can be obtained.

EXPERIMENT RESULTS

Evaluation

According the TREC authority, there are three metrics:

- Expected Gain. One way to evaluate an update system is to measure the expected gain for a system update. This is similar to traditional notions of precision in information retrieval evaluation.
- Comprehensiveness. Similar to tradition notions of recall in information retrieval evaluation.
- F measure. In order to summarize expected gain and comprehensiveness, we use an F measure based on both Expected Gain and Comprehensiveness.

Results

Table 1 shows the results of our system. In the first line of Table 1, EG signifies the scores of the expected gain, C signifies the scores of the comprehensiveness, F signifies the scores of F measure. In the second row, Q0 and Q1 is the runs we submitted, AVG is the mean score for each topic over all runs submitted to the track. In the first column of Table 1, the meaning of per-topic is obviously, mean signifies the average values of the scores over the 15 topics are given for each run. In the second column of Table1, All signifies the mean score over all topics and all runs submitted to the track.

Through Table 1, the performance of Q0 and Q1 with respect to the metrics Expected Gain and F measure are mostly better than AVG, which means that our methods are effectively. However, there are several topics whose Comprehensiveness value is smaller than the AVG, which means that our methods are not so well in recall. Through the contrast of the last three lines, we come to the conclusion that expect

Comprehensiveness, our run's performance is better than the average.

Conclusion

In this paper, we presented the implementation details of our runs for Temporal Summarization Track, and our runs performed well respect to Expected Gain and F score, but not so well respect to Comprehensiveness. The possible reason is that we excessive emphasis on the relevance between topic and sentence, and ignored the comprehensiveness of topic. Therefore, the future work's emphasis should be on how to improve the Comprehensiveness (or recall).

References

1. Z. Yang, F. Yao, H. Sun, Y, Zhao, BJUT at TREC 2013 Temporal Summarization Track, 2013
2. http://www.trec-ts.org/
3.http://s3.amazonaws.com/aws-publicdatasets/trec/ts/streamcorpus-2014-v0_3_0-ts-filtered/index.html
4. http://www.lemurproject.org
5. http://en.wikipedia.org/wiki/BM25
6. G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620, 1997
7. T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record, 25(2), 103-114, 1996.