# ISCASIR at TREC 2015 Temporal Summarization Track

Peixia Wang, Wenbo Li*
The National Engineering Research Center of Fundamental Software
The Institute of Software, Chinese Academy of Sciences (ISCAS)
Beijing, China
{peixia, wenbo}@.nfs.iscas.ac.cn

## Abstract

The goal of Temporal Summarization task is to develop systems which can detect useful, new, and timely sentence-length updates about a developing event. This paper describes our participation in Temporal Summarization track of TREC2015.

Based on the word embedding technique, we submitted two runs for the summarization task. The query expanding technique is used for the first run and relevant sentences are retrieved by computing the distance between the expanded query and the sentence. The processing of second run is the same with the first run except for the query expanding stage. Using the KBA Stream Corpus 2014, the experimental results show the effectiveness of our approach.

## 1 Introduction

As the Temporal Summarization 2015 Guidelines[1] describes, the goal of the TS track is to develop systems which can detect useful, new, and timely sentence-length updates about a developing event. There are three sub-tasks in TREC 2015, however, we only participate in the third sub-task because of the time limit, i.e. Task 3: Summarization Only. During the task, participants will be provided low-volume streams of on-topic documents for a set of topic events and it requires each participant to process those streams in time order, that is to say, the participant needs pick up relative sentences from the documents contained within each stream as updates over time.

## 2 Our Approach

The way to select relative sentences from the data stream is inspired by WMD distance[2] which use a new metric for the distance between text documents. Similarly, we measure the document distance by the cumulative amount of distance that the embedded query words of the topic event match the embedded words of the candidate sentence. The difference between our proposed approach and the WMD lies in the specific function of distance computation between sentences, details are described in the following part .

Our approach leverages recent results by Mikolov[3],i.e. word2vec model which we use to generate high-quality vector representations of words considering that it can

capture precise syntactic and semantic word relationships. A particular implementation of neural network based algorithm for training the word vectors is available at code.google.com/p/word2vec. After the training converges, words with semantic relevance are mapped into a similar space in the vector space and therefore we use the distributed representation of words to compute the distance between the query and the sentence.

In addition, it is necessary to preprocess the data stream to the format we would like to use.

## 2.1 Preliminaries

The corpus we use is the KBA Stream Corpus 2014, i.e. the second filtered set TREC-TS-2015F-RelOnly[4] that consists of a manually selected set of relevant documents for each event because we only participate Task 3. As the data inside each corpus file is encrypted and serialized with thrift format. So it is necessary to preprocess the corpus into the data format that is easy to deal with before we use it.

Firstly, decrypting the files uses the authorized key and converts the .GPG file format to .SC file format;

Secondly, deserialize the data into the sentence lists on demand by ways of interacting with stream corpus chunks using the tools provided by the streamcorpus project in github[6]. The preprocessing stage produces the processed corpus which is to be used in the next stage. The output format of the processed data is in the following tab-separated format:

Table 1: the format of the processed corpus

| | | | |
|---|---|---|---|
| 1358355262-78a6fa3abc32368d90f701cf69fbb885 | 1358355262 | 5 | Helicopter Crash In Vauxhall : Pilot Named |
| 1358355262-78a6fa3abc32368d90f701cf69fbb885 | 1358355262 | 6 | He died after the aircraft hit a crane on St George Wharf Tower , in Vauxhall , amid heavy fog . |
| 358355262-78a6fa3abc32368d90f701cf69fbb885 | 1358355262 | 7 | It cartwheeled out of the sky , smashed into two cars as it hit the ground and exploded into flames . |

where the columns are defined as,
The first column: document identifier
The second column: decision timestamp
The third column: sentence identifier
The fourth column: sentence content

## 2.2 Algorithm

We submitted two runs for the Task 3. The difference of the two runs lies in the ways of processing the query items. The first run is runvec1, parts of the query items are expanded and the second is runvec2, same with runvec1 except for the query expanding part.

The key point of expanding phase is to obtain a query item list by adding top $k$ words to the query items according to the semantic distance computed from the word vectors. Next, remove the stop words from the list with high frequency. In the end, add the event type to the newly query items due to its discriminate feature. Except for

the expanding stage, the processing progress of runvec2 is exactly the same with runvec1. The common processing parts of the two runs can be described as follows:

First, compute the cumulative similarity distance between the newly query items and the sentence items for every topic event.

Second, check whether the value of the distance is greater than the specified threshhold, if so, then check to see if the result sets contains the sentence, if not, add the sentence to the result sets.

The following is the pseudocode of algorithm:

**Algorithm 1:**
**Input:** stream of processed corpus
**Input:** topic queries
**Output:** list of sentence identifiers
1: Initialize: *RESULT*={}
2: **for** each query *q* **do**
3: expand the query q as *q*'
4: **for** each sentence *s* in processed corpus **do**
5: compute the *dist(q',s)*
6: if *dist(q',s)>threshhold* and *s* is not contained in *RESULT*
7: add *s* to *RESULT*
**8:** **end if**
**9:** **end for**
**10:** **end for**
14: return *RESULT*

*Where dist(q',s) is defined as:*

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} sim(W_i,W_j)}{\sqrt{n}} , \quad (1)$$

*m* is length of the query, *n* is length of sentence and $sim(W_i,W_j)$ is the similarity metric of the two words separately in query and sentence. The similarity can be obtained by computing the dot product of the corresponding two word vectors. Formerly, the distance between two words is defined as :

$$sim(W_i,W_j)=W_i.W_j, \quad (2)$$

Where $W_i$ is one word vector from query items and $W_j$ is one word vector from the sentence items.

Intuitively, the similarity between the query and the sentence can be represented by the matching degree of the corresponding sentences. The matching is measured by the cumulative distance that all word items in query match the words in the sentence. Furthermore, word matching is defined as the dot distance of the two word vectors.

As to sentence de-duplication, to avoid redundancy in updates and to improve the quality, duplicate sentences are forbidden to go into the result sets. We check whether the sentence exists in the result sets first, if so, delete the sentence and process next one.

## 3 Evaluation & Results

According the TREC authority, there are several metrics, such as (normalized) Expected Gain, Comprehensiveness and HM metric[5] and etc..

Expected Gain metric. It is the way to evaluate the relevance or precision of the summarization with respect to the event topic, something like the precision in traditional information retrieval.

Comprehensiveness metric. It is the way to measure the coverage of the summarization with respect to all the essential information contained in the corpus, similar to tradition concept of recall in information retrieval evaluation.

HM metric. A combined way to incorporate Expected Gain and Comprehensiveness with Latency included.

We submitted totally two runs for Task 3: runvec1 and runvec2. The results based on these metrics are as follows.

Table 2. The comparison of runvec1 and runvec2 for Task 3

| | | **nE[LG]** | | **Latency Comp.** | | **HM** | |
|---|---|---|---|---|---|---|---|
| Topic | ID | runvec1 | runvec2 | runvec1 | runvec2 | runvec1 | runvec2 |
| | 26 | 0.0096 | 0.0104 | 0.7116 | 0.7448 | 0.0190 | 0.0206 |
| | 27 | 0.0098 | 0.0176 | 0.2796 | 0.8049 | 0.0189 | 0.0345 |
| | 28 | 0.0038 | 0.0046 | 0.0046 | 0.2435 | 0.0075 | 0.0090 |
| | 29 | 0.0278 | 0.0299 | 0.5152 | 0.4701 | 0.0528 | 0.0563 |
| | 30 | 0.0166 | 0.0227 | 0.6154 | 0.5200 | 0.0323 | 0.0436 |
| | 31 | 0.0001 | 0.0108 | 0.0046 | 0.3372 | 0.0003 | 0.0208 |
| | 32 | 0.0153 | 0.0128 | 0.1740 | 0.1004 | 0.0281 | 0.0228 |
| | 33 | 0.0077 | 0.0078 | 0.3780 | 0.5093 | 0.0150 | 0.0153 |
| | 34 | 0.0134 | 0.0134 | 0.8969 | 0.8969 | 0.0263 | 0.0263 |
| | 35 | 0.0093 | 0.0093 | 0.6923 | 0.6923 | 0.0183 | 0.0183 |
| | 36 | 0.0104 | 0.0104 | 0.7750 | 0.7750 | 0.0205 | 0.0205 |
| | 37 | 0.0234 | 0.0223 | 0.5155 | 0.5461 | 0.0448 | 0.0429 |
| | 38 | 0.0138 | 0.0156 | 0.7012 | 0.7012 | 0.0271 | 0.0306 |
| | 39 | 0.0057 | 0.0067 | 0.7693 | 0.7693 | 0.0114 | 0.0134 |
| | 40 | 0.0078 | 0.0106 | 0.7288 | 0.6547 | 0.0154 | 0.0208 |
| | 41 | 0.0044 | 0.0041 | 0.2995 | 0.3619 | 0.0086 | 0.0080 |
| | 42 | 0.0077 | 0.0109 | 0.6020 | 0.5877 | 0.0151 | 0.0213 |
| | 43 | 0.0088 | 0.0082 | 0.7822 | 0.7822 | 0.0174 | 0.0162 |
| | 44 | 0.0202 | 0.0239 | 0.7690 | 0.7690 | 0.0394 | 0.0463 |
| | 45 | 0.0117 | 0.0137 | 0.6987 | 0.6987 | 0.0229 | 0.0269 |
| | 46 | 0.0050 | 0.0055 | 0.2426 | 0.2426 | 0.0099 | 0.0107 |

The table 2 above shows the detail on three metrics with latency of each topic and the table 3 shows the extended comparison of our two submitted runs: runvec1 and runvec2.

Table 3. The extended comparison of runvec1 and runvec2 for Task 3

| Run ID | | nE[Gain] | nE[Latency Gain] | Comp. | Latency Comp. | HM |
|---|---|---|---|---|---|---|
| runvec1 | STD | 0.0100 | 0.0066 | 0.1962 | 0.2457 | 0.0125 |
| | MIN | 0.0033 | 0.0001 | 0.1163 | 0.0046 | 0.0003 |
| | MAX | 0.0421 | 0.0278 | 0.9844 | 0.8969 | 0.0528 |
| | AVG | 0.0174 | 0.0111 | 0.7852 | 0.5409 | 0.0215 |
| runvec2 | STD | 0.0112 | 0.0067 | 0.1649 | 0.2139 | 0.0128 |
| | MIN | 0.0076 | 0.0041 | 0.3767 | 0.1004 | 0.0080 |
| | MAX | 0.0520 | 0.0299 | 0.9793 | 0.8969 | 0.0563 |
| | AVG | 0.0190 | 0.0129 | 0.7881 | 0.5813 | 0.0250 |
| ALL | AVG | 0.0595 | 0.0319 | 0.5627 | 0.3603 | 0.0472 |

From the table, we can conclude that the results show the effectiveness of our method in terms of recall and that we manage to retrieve most of the relevant updates covering the important nuggets, but the precision is lower than average. Furthermore, the expanding technique on the query items does not improve the precision and recall. Besides, the values of metrics fluctuate violently between the minimum and maximum for different event topics on which should be improved in the future.

## 4 Conclusion

This paper reports a word embedding-based framework and technical scheme for Task 3 in TREC 2015 Temporal Summarization Track. The soul of method is to get the distributed representation of words first and use it later to get the relative sentences with respect to the topic event. In addition, filtering out duplicate sentences is important too. This year, we do research on the existed word embedding only. In the future, we will take consider in more information on the embedding ways of sentences and Knowledge Base.

## 5 Acknowledgements

We would like to thank all organizers and assessors of TREC and NIST. This work was supported by the National High Technology Research and Development 863 Program of China under Grants no. 2013AA01A603.

## 6 Reference

[1] Aslam, Diaz, Ekstrand-Abueg, McCreadie, Pavlu, Sakai.Temporal Summarization. Available: https://38309103-a-62cb3a1a-s-sites.googlegroups.com/site/temporalsummarization/trec2015-ts-guidelines-updated.pdf.

[2] Matt J. Kusner , Sun Y. , Nicholas I. Kolkin , Kilian Q. Weinberger .From Word Embeddings to Document Distances. Proceedings of the 32nd International

Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.
[3]Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S.,and Dean, J. Distributed representations of words and phrases and their compositionality. In NIPS, pp. 3111–3119, 2013b.
[4] TREC-TS-2015F-RelOnly dataset. Available: http://dcs.gla.ac.uk/~richardm/TREC-TS-2015RelOnly.aws.list
[5] streamcorpus project. Available:https://github.com/trec-kba/streamcorpus/
[6] Aslam, Diaz, Ekstrand-Abueg, Pavlu, Sakai. Metrics.