ZZISTI at TREC2013 Temporal Summarization Track

Yaoyi Xi, Bicheng Li, Jie Zhou, Yongwang Tang
Zhengzhou Information Science and Technology Institute
Zhengzhou, P.R. China, 450002
wim.gy2013@gmail.com

Abstract

Our team submitted runs for the first running of the TREC Temporal Summarization track. TS Track at TREC2013 contains two tasks, namely Sequential update Summarization and value tracking. Our Systems to each task are described in this paper respectively. In particular, Stanford CoreNLP was applied to extract the event attributes.

1. Introduction

The goal of the Temporal Summarization track is to develop systems that allow users to efficiently monitor the information associated with an event over time. It focuses on two tasks, sequential update summarization and value tracking. The former requires broadcasting useful, new, and timely sentence-length updates about a developing event, while the latter needs to track the value of important event-related attributes (e.g. number of fatalities, financial impact).

Document summarization technique is a hot research topic in recent years, such as single and multi-document summarization, update summarization and so on. TS differs from previous summarization techniques in two primary ways: it is oriented to an online, sequential setting, and it needs to extract and track the value of important event-related attributes in dynamic settings.

TS Track at TREC2013 used the TREC KBA 2013 Stream Corpus. This corpus consists of a set of times-tamped documents from a variety of news and social media sources covering the time period October 2011 through January 2013. A document contains a set of sentences, each with a unique identifier, which is the index of the sentence in the document, beginning at zero. For the purpose of the TS track, the corpus of time-stamped

documents is considered a stream and documents should be iterated over in temporal order. We have used the KBA 2013 ‘English-and-unknown-language’ streamcorpus with all non-English documents removed and the StreamItem.body.raw text set to "". This stripped corpus is about 4.5TB and just over 500M StreamItems.

The remainder of this paper is organized as follows: Section 2 on our sequential update summarization system and Section 3 on our value tracking system. Section 4 shows the results of our submitted runs. We conclude in Section 4.

2. Sequential Update Summarization

According to the sequential update summarization task, a system should emit relevant, important and novel sentences to an event. We submitted one run for the sequential update summarization task, and denote it as SUS1 in the following paper. The implementation of SUS1 is shown in Figure1. We described each implementation step respectively in section 2.1, 2.2, 2.3 and 2.4. Section 2.5 gives the other strategies in SUS1.

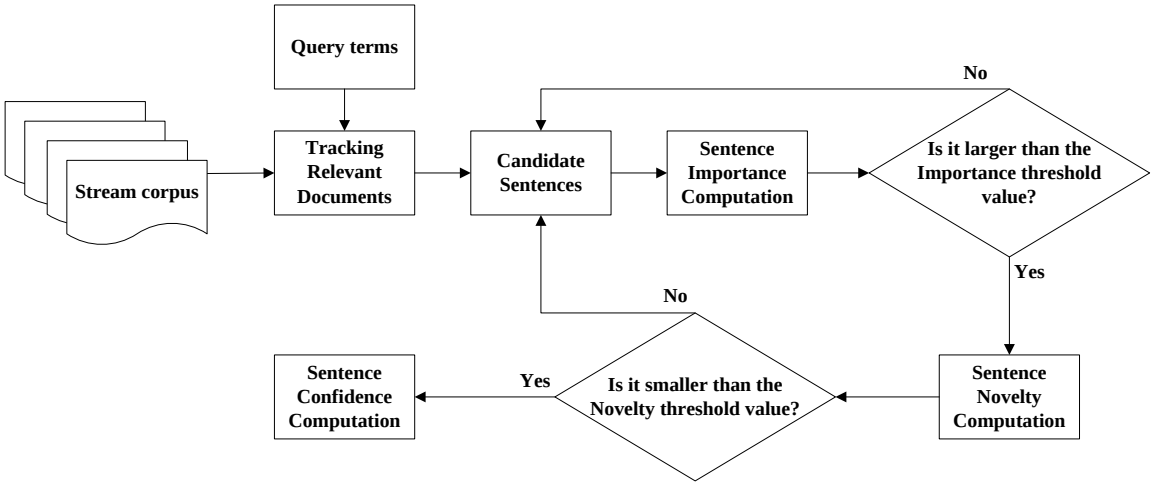

Figure 1 The implementation of SUS1

2.1 Tracking Relevant Documents

The summary sentence must be from the document which is relevant to the topic. To quickly find these documents from the extremely large stream corpus, we assume that one document is about the topic if it contains all the query terms. In addition, to help

speed up processing, SUS1 discarded the documents containing more than K sentences with the assumption that relevant documents usually do not have too much sentences. In TS2013, we set $K = 40$.

2.2 Sentence Importance Computation

The summary sentence certainly contains important information. Based on a lot of observation results, we have found that summary sentences usually have more entities than others and the entities generally contain important topic information. Therefore, we assume that the number of entities in a sentence reflects the amount of information in it and we define the sentence importance score as follows:

$$SenIMP(s,d) = \alpha \times \frac{\sum_{t \in s} W(t,d)}{usefulTokenNum} \quad (1)$$

where $SenIMP(s,d)$ denotes the importance score of sentence s in document d, $W(t,d)$ defines the weight of term t in document d, $usefulTokenNum$ denotes the number of entities in sentence s, the parameter α controls the degree of the prize when sentence s contains some of the query terms. We set $\alpha = 1$ when s doesn't have any of the query terms, and $\alpha = 1.5$ when s contains some of the query terms. $W(t,d)$ is defined as:

$$W(t,d) = \frac{tf(t,d) \times \log(N / n_t)}{\sqrt{\sum_{t \in d} [tf(t,d) \times \log(N / n_t)]^2}} \quad (2)$$

where $tf(t,d)$ denotes the frequency of term t appears in document d, N denotes the number of training documents, n_t denotes the number of documents in the training corpus that contain term t. If term t doesn't appear in the training corpus, we assume that $n_t = 2$. To avoid using "future" data, we construct the training corpus by choosing documents from the news substream in the stream corpus, ranging from November 1 to November 2, 2011.

2.3 Sentence Novelty Computation

We intend to measure the novelty of a sentence by computing the similarities between it

and the summary sentences existed. We define the novelty of a sentence as:

$$Novelty(s,d) = \frac{\sum_{s_e \in U} Cos(s, s_e)}{M} \quad (3)$$

where U defines the set of the summary sentences existed, s_e denotes the summary sentence in U, M denotes the size of U, $Cos(s, s_e)$ denotes the cosine similarity between s and s_e. The novelty of sentence s decreases with $Novelty(s,d)$ increasing.

2.4 Sentence Confidence Computation

TS2013 requires the simulated system to give a confidence value for each update. This value encodes the system's confidence in this being a reasonable update, which may be used to prioritize updates if the assessors cannot judge all of the updates. We define the sentence confidence value as follows:

$$CS(s,d) = rIMP(s,d) \times rNov(s,d) \quad (4)$$

where $CS(s,d)$ denotes the confidence value of sentence s in document d, $rIMP(s,d)$ denotes the relative importance of sentence s, and $rNov(s,d)$ denotes the relative novelty of sentence s. The calculation of $rIMP(s,d)$ and $rNov(s,d)$ are given respectively:

$$rIMP(s,d) = \frac{SenIMP(s,d) - \theta}{SenIMP(s,d)} \quad (5)$$

$$rNov(s,d) = \frac{\varepsilon - Novelty(s,d)}{\varepsilon} \quad (6)$$

where θ denotes the importance threshold, ε denotes the novelty threshold.

2.5 Other Strategies

Strategy1: We judge the novelty of a sentence by computing its similarity with the existing summary sentences. Therefore, the selection of the first N summary sentences is very important. If they were chosen inaccurately, the accuracy of the subsequent sentence selection would be affected. Since the query terms are usually the most

representative ones in a topic, we assume that the first N summary sentences must contain some of the query terms. In TS2013, we set $N = 3$.

Strategy2: By observed, we found that the sentences containing important information usually have an appropriate length, neither too long nor too short. So we filter out the sentences which length are greater than L_1 or less than L_2. In TS2013, we set $L_1 = 40$, and $L_2 = 10$.

3 Value Tracking

According to the value tracking task, a system should emit accurate attribute value estimates for an event. This is analogous to extracting argument roles for a given event template. The only difference is that the value tracking task just focuses on specific event arguments, including Deaths, Injuries, Displaced, Financial Impact and Locations. For all that, we followed the rule-based method in our value tracking systems. We have used the training event data and the stream corpus that existed before the test events when we generated the extraction rules.

We submitted 2 runs for the value tracking task. Each run is described in detail in the following paper.

3.1 Value Tracking Run1

For simplicity, we denote our first value tracking system as VT1 in the following paper. In VT1, we firstly tracked the relevant documents as section 2.1. Secondly, we filtered out the noise sentences from the relevant documents. Here, the noise sentence refers to the sentence that doesn't have the event attributes. The specific filtering rules are shown in Figure 2. Thirdly, we extracted the desired event attributes from the remnant sentences using the extraction rules, and set the initial confidence value of the attribute at 0.5. Finally, we refined the confidence value by weighting it according to the sentence importance and the number of attributes in this sentence.

1. Filtering out the sentence that is shorter than 10 words in length;
2. Filtering out the sentence which importance value is lower than 0.11;
3. Filtering out the sentence that doesn't have any of the query terms.

Figure 2 The filtering rules of VT1

3.2 Value Tracking Run2

There are some limitations of the rule-based method, for example, the attribute extracted by this method may be not an event attribute at all. To improve the rule-based method, we attempt to use the Stanford CoreNLP to recognize the named entities in the sentence, such as NUMBER, MONEY, LOCATION and so on. We think the Deaths, Injuries and Displaced attributes belong to NUMBER, Financial Impact attributes belong to MONEY, and Locations attributes belong to LOCATION. Based on this, we developed the second value tracking system, VT2. The main idea of VT2 is using CoreNLP to validate the attribute extracted by the rule-based method. If the extracted results of the two methods are the same, then we retain this attribute. If not, we discarded it. The process flow of VT2 is:

Firstly, tracking the relevant documents;

Secondly, filtering out the noise sentences;

Thirdly, extracting the event attributes from the remaining sentences with the rule-based method and setting the initial confidence value of the attributes at 0.5;

Fourthly, using CoreNLP to extract named entities from the sentences. If the extracted results are the same as the ones specified in Step3, the confidence value of the attributes is increased to 0.75;

Finally, refining the attributes' confidence value by weighting them according to the sentence importance and the number of attributes in this sentence. Other than the forth step, any other steps above are the same as in VT1.

The important thing to note here is the version of CoreNLP. TS2013 requires that external data must have existed before the event start time, or be time-aligned with the KBA corpus and no information after the simulation decision time can be used. So we choose Version 1.3.0 in VT2, which was released on January 8, 2012 by the Stanford Natural Language Processing Group.

4. Evaluation

In TS track2013, 26 submissions were made to the sequential update summarization task, and 7 submissions were made to the value tracking task. Among which, our team contributed 1 submission to the sequential update summarization task and 2 submissions to the value tracking task.

4.1 Data Set and Evaluation Metrics

This year's Temporal Summarization track contained 10 topics. They include 2 accidents, 2 shootings, 4 storms, 1 earthquake, and 1 bombing. For each of these topics, the summarization time window is 10 days.
Some metrics were developed by the track organizers to measure the quality of runs. For Sequential Update Summarization, Expected Latency Gain (ELG) and Latency Comprehensiveness(LC) of each run were used. For Value Tracking, Expected Error(EE) was used.

4.2 Result Analysis

Table 1 and 2 report the performance of our submitted run for the sequential update summarization task in terms of expected latency gain and latency comprehensiveness respectively. Topic 7 happened in early July, but the streamcorpus doesn't have data for that time period, other than arxiv, so the organizers have ignored this topic.

	1	2	3	4	5	6	8	9	10
SUS1	0.072	0.070	0.006	0.029	0.000	0.003	0.045	0.022	0.074
AVG	0.077	0.065	0.099	0.073	0.002	0.040	0.053	0.038	0.091
MAX	0.278	0.186	0.425	0.284	0.010	0.160	0.099	0.090	0.265

Table 1 Expected Latency Gain of SUS1 Over 10 Evaluation Topics

	1	2	3	4	5	6	8	9	10
SUS1	0.390	0.230	0.004	0.069	0.000	0.002	0.099	0.141	0.396
AVG	0.562	0.327	0.114	0.239	0.009	0.051	0.294	0.426	0.676
MAX	0.990	0.630	0.342	0.514	0.030	0.120	0.608	0.999	0.996

Table 2 Latency Comprehensiveness of SUS1 Over 10 Evaluation Topics

The performance of our run with respect to the ELG and LC metric, are below the average reported amongst all submitted runs to the track. This could be in part because we have used 'StreamItem.body'[1] to filter the streamcorpus, but it is always incomplete, which led to many relevant documents not included. On the other hand, only use the named entities to measure the importance of sentence may be not appropriate.
Although the general performance, SUS1 does not take use of any other external resources and is easy to implement. It may be used as a baseline in the future.
Table 3 reports the performance of our submitted runs for the value tracking task, VT1 and VT2, in terms of expected error. TREC2013 did not give the performance of the

attribute “Displaced” in all runs. One possible reason is that all runs performed poorly in extracting the displaced attribute values.

	location	deaths	injuries	financial impact(10^9)
baseline	20038.0	195.111	473.222	13.3539
VT1	14483.6	2726.06	410.092	13.3539
VT2	4660.76	2396.12	410.531	13.3539
AVG	14426.08	28172.55	39863.62	18.65641
MIN	4660.76	138.1	390.985	9.5251

Table 3 Expected Error of VT1 and VT2 Over 10 Evaluation Topics

Table 3 shows that low expected error is achieved in VT1 and VT2, which can be attributed to the effectiveness of the extraction rules. In particular, VT2 performed better than VT1, which confirmed that using of Stanford CoreNLP can improve the extraction performance.

5. Conclusion

In this paper, we presented the implementation details of our runs for the Temporal Summarization Track. The TS Track is a very challenging task as expected and therefore very interesting. This first year allows us to comprehend what is behind TS. Overall, none of the submitted runs performed well both in expected latency gain and latency comprehensiveness, and there are still many improvements that can be done. Since this is only the first year, it will make the following years quite promising.

References

[1] TREC Temporal Summarization. http://www.trec-ts.org/, 2013.