IRIT at TREC Temporal Summarization 2014

Rafik Abbes, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem

{abbes, sauvagnat, hernandez, boughanem}@irit.fr,
IRIT, Paul Sabatier University
118 route de Narbonne F-31062 Toulouse cedex 9

Abstract. This paper describes the IRIT lab participation to the 2014 TREC Temporal Summarization track. The goal of the Temporal Summarization track is to develop systems that allow users to efficiently monitor information about events over time. Our proposed method selects relevant documents that are more likely to concern the event, and extracts relevant and novel sentences based on some filters. Obtained results are presented and discussed.

1 Presentation of the task

The aim of the Temporal Summarization (TS) track is to develop systems that allow users to efficiently monitor information about events. This year, the track run only one task which requires systems to iterate over a stream corpus in a chronological order and filter relevant and novel sentences to a developing event.

A specially filtered subset of the full TREC 2014 StreamCorpus[1] was provided. It consists of about 20 million documents from several sources (News, Social, Forum, Blog, etc.) having a size of 559 GB (compressed). Each document is identified by a *stream_id* that consists of two dash-separated parts: *timestamp* and *doc_id*. This year, 15 topics were evaluated. Each topic represents an event characterized by a *title*, a *Wikipedia URL*, a *period*, a *query* and a *type* (accident, storm, bombing, riot, protest, impact event, shooting). For each event, a system should emit a set of timestamped sentences called *updates* to generate the event summary. Ground truth, called *nuggets*, corresponds to a set of sentences extracted from Wikipedia by the track annotators. Matching updates to nuggets was done by track assessors. A nugget and an update are matched if they refer to the same information. To evaluate systems effectiveness, track organizers define two metrics: the *Expected Latency Gain* (ELG) and the *Latency Comprehensiveness* (LC) which are similar to the traditional IR notions of Precision and Recall (respectively). Systems are ranked based on the harmonic mean between ELG and LC.

2 IRIT method for temporal summarization

Our system is based on Algorithm 1 which is similar to the one given in the track guidelines[2], used as reference in some methods at TREC TS 2013 [1, 2]. Given an event query Q_e and its corresponding period (start time t_s, end time t_e), our system iterates over the stream corpus in a chronological order, hour by hour (line 2 of the algorithm).

We can distinguish 3 basic steps: the first one, done just once, build a generic event model containing a bag of weighted terms related to events (line 1 of the algorithm). Step 2 and 3 are repeated iteratively for each hour. In step 2, our system has to decide which documents

[1] http://s3.amazonaws.com/aws-publicdatasets/trec/ts/index.html
[2] http://www.trec-ts.org/documents

Algorithm 1 Temporal Summarization algorithm

Input: C : Time-ordered corpus
Input: Q_e : Event query terms
Input: t_s : Event start time
Input: t_e : Event end time
Input: N_{train} : Set of training nuggets
Output: $U \leftarrow \{\}$

1: $\theta_E \leftarrow BuildGenericEventModel(N_{train})$
2: **for** $h \in [toHour(t_s), toHour(t_e)]$ **do**
3: $\quad D_h \leftarrow getRelevantDocuments(h, Q_e, topHits)$
4: $\quad$ **for** $d \in D_h$ **do**
5: $\quad\quad$ **for** $s \in d$ **do**
6: $\quad\quad\quad$ **if** $isUsefulSentence(s, U)$ **then**
7: $\quad\quad\quad\quad U.append(u)$
8: $\quad\quad\quad$ **end if**
9: $\quad\quad$ **end for**
10: $\quad$ **end for**
11: **end for**

should be kept in order to extract updates (line 3 of the algorithm), and in the last step, it attempts to detect relevant and novel sentences related to the event (line 6 of the algorithm). These steps are detailed below.

2.1 Generic event model

We hypothesize that updates related to events tend to contain a specific vocabulary of terms independent of the event type (storm, hurricane, bombing, etc.) such as *victims*, *injuries*, *deaths*, *emergency*, etc. We call these terms *keywords*. We assume that we can build a generic event model by leveraging a set of nuggets related to a sample of events. Specifically, considering a set of training nuggets N_{train} related to m events, we estimate the generic event model θ_E composed of terms t as follows:

$$P(t|\theta_E) = \frac{TF(t, N_{train})}{log(\frac{m}{EF(t)})}$$

Where $TF(t, N_{train})$ is the term frequency of term t in the training nuggets N_{train}. $EF(t)$ represents the number of events containing term t in at least one nugget. Thus, a term is more weighted when it appears in most of the training events.

2.2 Document selection

To reject documents that are more likely to be not relevant, we apply the following filters:

- *Source filter*: Based on some analysis done on the last year's results (TS 2013), we noticed that 95% of relevant document (i.e., having at least one relevant sentence) come from one of the following sources: *WEBLOG*, *MAINSTREAM_NEWS* and *news*. For this reason, we reject all documents coming from other sources.
- *Title filter*: 95% of relevant document in TS2013 have a title. We therefore reject documents without titles.
- *Duplication filter*: We reject also duplicate documents having the same *doc_id*.

In each hour, we keep only the **topHits** filtered documents based on the following score:

$$Score(d, e) = \sum_{t \in Q_e} \frac{TF(t, d)}{|d|}$$

Where $TF(t, d)$ is the number of occurrences of term t in document d.

2.3 Sentence selection

In this step, our system parses all sentences of the selected documents. A sentence is worthy to be added to the summary of the event if it fulfills all the following conditions:

- not too verbose, i.e., it contains less than 25 words.
- it matches at least one query term (one term from Q_e).
- it matches at least one keyword from the generic event model θ_E.
- it is novel, i.e. not similar to recently added sentences in the last *novelty window* denoted by ***NW***. We consider that sentences S1 and S2 are similar if $cos(S1, S2) > 0.5$.

The score of each selected sentence is then the sum of weights of keywords.

3 Submitted runs and results

In our submitted runs, we used the nuggets of TREC TS 2013 as training data to build the generic event model θ_E. In addition, as shown in Table 1, we evaluated different values of *number of keywords*, *top documents per hour* and *novelty window*.

- We evaluated two sets of ***keywords***: a small set consisting of the top-30 terms in θ_E, and a larger one containing the top-80 terms of θ_E.
- We evaluated two values of ***topHits*** (top documents per hour): 5 and 10.
- We measured the novelty score of a sentence considering the two values of novelty window ***NW***: 300 sec. (5 min.) and 3600s (1 hour).

run label	#keywords	topHits	NW (sec.)
KW30H10NW300	30	10	300
KW30H5NW300	30	5	300
KW30H5NW3600	30	5	3600
KW80H10NW300	80	10	300
KW80H5NW300	80	5	300
KW80H5NW3600	80	5	3600

Table 1: Our different configurations.

Results are shown in Table 2. We notice that the best results are obtained by considering the top-5 documents in each hour and 30 keywords. In fact, considering the top-10 documents per hour improves the LC (recall) but brings much noise which degrades the ELG (precision) of the system. Moreover, selecting sentences based on the small set of keywords (30 terms) seems to be enough to get a good recall. However, further work are needed to refine the precision of the summarization process. Concerning the novelty detection, considering 5 *minutes* as novelty window seems to be too short to detect redundant updates.

	ELG	***LC***	***H***
KW30H10NW300	0.0348	0.4838	0.0602
KW30H5NW300	0.0429	0.4323	0.0714
KW30H5NW3600	**0.0434**	0.4315	**0.0723**
KW80H10NW300	0.0279	**0.5199**	0.0503
KW80H5NW300	0.0339	0.4704	0.0596
KW80H5NW3600	0.0344	0.4679	0.0604

Table 2: Results of our different configurations. H is the harmonic mean between ELG and LC over all events.

References

1. Qian Liu, Yue Liu, Dayong Wu, and Xueqi Cheng. ICTNET at temporal summarization track TREC 2013. In *Text REtrieval Conference (TREC)*, 2013.
2. Yaoyi Xi, Bicheng Li, Jie Zhou, and Yongwang Tang. ZZISTI at TREC 2013 temporal summarization. In *Text REtrieval Conference (TREC)*, 2013.