TREC 2015 Temporal Summarization

Temporal Summarization 2015 Guidelines

August 24, 2015 Aslam, Diaz, Ekstrand-Abueg, McCreadie, Pavlu, Sakai

1 Introduction

There are many summarization scenarios that require updates to be issued to users over time. For example, during unexpected news events such as natural disasters or mass protests new information rapidly emerges. The TREC Temporal Summarization track aims to investigate how to effectively summarize these types of event in real-time. In particular, the goal is to develop systems which can detect useful, new, and timely sentence-length updates about a developing event. There are two sub-tasks running in 2015:

Task 1: Filtering and Summarization

- Participants will be provided *high-volume streams* of news articles and blog posts crawled from the Web (TREC-TS-2015 a.k.a. KBA Stream Corpus 2014).
- Each participant will need to process those streams in time order, *filter out irrelevant content* and then select sentences from those documents to return to the user as updates describing each event over time.

Task 2: Pre-Filtered Summarization

- Participants will be provided pre-filtered *high-volume streams* of news articles and blog posts crawled from the Web for a set of events (TREC-TS-2015F).
- Each participant will need to process those streams in time order, *filter out irrelevant content* and then select sentences from those documents to return to the user as updates describing each event over time.

Task 3: Summarization Only

- Participants will be provided low-volume streams of *on-topic documents* for a set of events (TREC-TS-2015F-RelOnly).
- Each participant will need to process those streams in time order selecting sentences from the documents contained within each stream to return the user as updates over time.

In contrast to classical summarization challenges (such as DUC or TAC), the summaries produced by the participant systems are evaluated against a ground truth list of information nuggets representing the space of information that a user might want to know about each event. An optimal summary will cover all of the information nuggets in the minimum number of sentences.

2 Definitions

2.1 Corpus

We will be using the TREC KBA 2014 Stream Corpus. This corpus consists of a set of timestamped documents from a variety of news and social media sources covering the time period October 2011 through April 2013. A document contains a set of sentences, each with a unique identifier. Participants should get access to the corpus by submitting the appropriate KBA User Agreements. You may use the KBA 2014 'english-and-unknown-language' streamcorpus.

Due to the size and computational cost of processing the full KBA 2014 corpus, we will also be releasing two smaller versions that have been pre-filtered to focus on documents that are likely to contain relevant sentences. Which of these two corpus versions you use depends on which task(s) you are participating in. If you are participating in *Task 1: Filtering and Summarization*, you will be using the first filtered set, denoted `TREC-TS-2015F` that consists of the top documents from a high precision retrieval for each event. This corpus is still high-volume in nature and contains much irrelevant content, hence the filtering aspect of Task 1. If you are participating in *Tash 2: Summarization Only*, then you will be using the second filtered set `TREC-TS-2015F-RelOnly` that consists of a manually selected set of relevant documents for each event that you are to summarize. You may only use the TREC-TS-2015F-RelOnly version when submitting to Task 2, not Task 1.

2.1.1 Retrievable Units

The aim of both tasks is to *extract relevant, novel and informative sentences from the document stream* to return to the user as updates over time, forming an event summary. Participants should return a list of sentences from the KBA corpus for each event in the format detailed later in Section 3. Each sentence is identified by the combination of a document identifier and a sentence identifier. Documents in the KBA corpus are segmented into sentences in the 'ner' field. A sentence identifier is the index of the sentence in the document, *beginning at zero*. If there are three sentences in a document, the first sentence would be identified as '0', the second would be identified as '1', and the third would be identified as '2'. If there is only one sentence in a document, it would be identified as '0'. **It is your responsibility to make sure your output format is consistent with this indexing.**

Note: The KBA 2014 corpus using *two* sentence lists, you should start off trying to parse with 'serif', if the sentence list does not exist for that, then fall back to 'lingpipe'.

2.1.2 Processing Order

Temporal Summarization track aims to investigate the summarisation of events over time in as realistic a setting as possible. As such, participant systems should treat the documents from within the time period of each event as a stream. Therefore, documents should be iterated over in temporal order. Documents should be sorted by `stream_time.zulu_timestamp` (equivalently `stream_time.epoch_ticks`). We encourage participants to use efficient data structures to perform many experiments (i.e. 're-running the simulation'). **However, any data structure should not expose the simulated system to information with a timestamp after the decision time.** This means that participants should be careful about precomputing temporally sensitive global statistics such as IDF, a value which will change as documents are processed.

When emitting a sentence into the summary, a timestamp should be provided indicating when (with respect to the underlying document stream) the sentence was emitted. In the case of a fully real-time system that makes include/exclude decisions on a per-document basis, this timestamp would correspond to timestamp of the document containing each sentence. Alternatively, a system that buffers documents for a period of time before emitting (e.g. emitting at the end of each hour for instance) should produce a timestamp indicating when each batch of sentences were emitted with respect to the underlying document stream (e.g. using the **end of hour** if emitting updates for an hour's worth of buffered documents).

One alternative is to treat timestamps within the corpus' hour directory as the end of that hour and process all documents in the hour directory together (i.e. without the need for sorting). If your algorithm uses data across hour directories, you still must iterate over the hour directories in time order.

2.1.3 External Resources

Participants are allowed to include runs that use information external to the KBA corpus. We stress the following requirements,

- external data must have existed before the event start time, or
- external data must be time-aligned with the KBA corpus **and** no information after the simulation decision time can be used.

Similarly, supporting statistical models or auxiliary programs are subject to the same requirements. For example, participants should **not** use a statistical model trained on data that existed **after** the event end time.

```
<event>
  <id>1</id>
  <title>2012 Buenos Aires rail disaster</title>
  <description>http://en.wikipedia.org/wiki/2012_Buenos_Aires_rail_disaster</description>
  <start>1329910380</start>
  <end>1330774380</end>
  <query>buenos aires train crash</query>
  <type>accident</type>
</event>
```

Figure 1: Complete topic definition for ‘2012 Buenos Aires Rail Disaster’.

```
<event>
  <id>1</id>
  <start>1329910380</start>
  <end>1330774380</end>
  <query>buenos aires train crash</query>
  <type>accident</type>
</event>
```

Figure 2: Masked topic definition for ‘2012 Buenos Aires Rail Disaster’.

When submitting, any external data used should be declared. Unlike for the 2013 task, it is not required to submit a ‘basic’ run without such external data.

2.2 Topics

An event refers to a temporally acute topic and is represented as,

- **Title**: A short retrospective description of the event (string).
- **Description**: A retrospective free text event description (url).
- **Start**: Time when the system should start summarization (UNIX timestamp in GMT).
- **End**: Time when the system should end summarization (UNIX timestamp in GMT).
- **Query**: A keyword representation of the event description expressed by a user during the event (string).
- **Type**: The type of event (one of {accident, bombing, conflict, earthquake, hostage, impact event, protest, riot, shooting, storm}).

We present an example topic in Figure 1. We will provide participants with a set of masked event topics containing only the topic id, query, start, and end (Figure 2).

3 Task Definition

As discussed in the introduction, there are two tasks within the 2015 track:

- Task 1: Filtering and Summarization
- Task 2: Summarization Only

For both of these tasks, for an event, a participant system will process a stream of documents in time order and extract sentences to return to a user following that event as updates. Updates returned should be relevant, be informative, and contain new information with respect to what has been returned previously. Furthermore, updates that contain out-of-date information will be considered less valuable than those containing timely information. The key difference between the two tasks is the assumptions that the system can make with respect to the underlying document stream:

- For Task 1, most of the documents will be irrelevant, simulating a realistic deployment setting. Participants will first need to find the relevant documents from the stream and then extract informative sentences from those documents to return to the user.
- For Task 2, all documents within the stream are guaranteed to contain one or more relevant sentences about the event. Participants need only to extract informative sentences from those documents to return to the user. However, note many sentences within each document will be off-topic, as the KBA stream-corpus documents contain the boiler-plate content (e.g. navigation bars, links to other articles, etc.).

During the simulation, a system should emit relevant and novel sentences to an event (exact metrics will be released in a separate document). Conceptually, a simulator should be structured as in Figure 3. The arguments to the simulator are the participant summarization system, the time-ordered corpus, the keyword query, and the relevant time range. In line 1, we initialize the output summary to empty. In line 2, we initialize the sequential update summarization system with the event query. The system should store some representation of this query for later processing and filtering. We iterate over the corpus in temporal order (line 3), processing each document in sequence (line 5). If the document we are processing is in the event timeframe (line 7), then we check to see if adding the document resulted in the system deciding to output a set of summary sentence ids (line 9). We then add these sentence ids to the summary timestamped with the time of the decision (lines 10-12).

We have tried to present an abstract representation of sequential update summarization. There are several comments worth making. First, if a participant is interested in efficiency and does not anticipate needing documents outside of the event timeframe, then the call to S.PROCESS(d) can be moved inside of the condition in line 7. Participants should be clear about any filtering of $\mathcal{C}$. For example, a participant should note if they are just iterating over documents with a high BM25 score. However, if this is done, care must be taken to make sure that filtering out a document d does not exploit information from sources after d.TIME() (e.g. retrospective IDF values).

4 Result Format

We expect team result formats to be in the following tab-separated file format,

```
 1 HelloWorldUniversity  Cluster1  1357052200-54f6f6e096a4cae27bee55dc2e0dc2b6 0 1330432283 0.90
 1 HelloWorldUniversity  Cluster1  1357052200-54f6f6e096a4cae27bee55dc2e0dc2b6 1 1330472283 0.75
 1 HelloWorldUniversity  Cluster1  1357052190-8f7a8b30a9a8f671dcc979677b04fab4 1 1330482283 0.99
```

where the columns are defined as,

1. query identifier
2. team identifier
3. run identifier
4. document identifier
5. sentence identifier: base 0 index of a segmented sentence in the KBA corpus
6. decision timestamp
7. confidence value: a strictly positive number (> 0) which encodes the system's confidence in this being a reasonable update; this value may be used to prioritize updates if we cannot judge all of them.

For 2015, each run may contain at most 1000 sentences per event.

5 Important Dates

June 9, 2015	guidelines and metrics released
June 9, 2015	train and test events released
June 9, 2015	Task 1: TREC-TS-2015F corpus released
July 1, 2015	Task 2: TREC-TS-2015F-RelOnly released
late July 2015	submission website open
September 3, 2015	submission website closed
late November 2015	TREC Conference

```
TemporalSummarization($\mathbf{S}, \mathcal{C}, q, t_s, t_e$)
    $\mathbf{S}$      ▷ Participant system.
    $\mathcal{C}$     ▷ Time-ordered corpus.
    $q$       ▷ Event keyword query.
    $t_s$     ▷ Event start time.
    $t_e$     ▷ Event end time.
    $\mathcal{U} \leftarrow \{\}$
    $\mathbf{S}$.Initialize($q$)
    for $d \in \mathcal{C}$
        do
            $\mathbf{S}$.Process($d$)
            $t \leftarrow d$.Time()
            if $t \in [t_s, t_e]$
              then
                  $\mathcal{U}_t \leftarrow \mathbf{S}$.Decide()
                  for $u \in \mathcal{U}_t$
                      do
                          $\mathcal{U}$.Append($u, t$)
    return $\mathcal{U}$
```

Figure 3: Sequential update summarization simulator.