

## Draft Guidelines

April 22, 2013

# 1 Definitions

## 1.1 Corpus

We will be using the [TREC KBA 2013 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 January 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 2013 ‘english-and-unknown-language’ streamcorpus.

### 1.1.1 Retrievable Units

In addition to a document identifier, participants should always return a sentence identifier for each update. 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.**

### 1.1.2 Processing Order

For the purpose of the Temporal Summarization track, the corpus of timestamped documents will be considered a stream. Therefore, documents should be iterated over in temporal order. 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 indexed.

### 1.1.3 External Resources

For 2013, participants are *not* allowed to use any information outside of the evaluation corpus in official runs. We are focusing on the core task and metrics this year but plan on revisiting this question in subsequent years.

## 1.2 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).
- **Location:** The location where the event happened (list of (latitude,longitude,time) values).
- **Dead:** The number of deaths related to the event (list of (count,time) values).
- **Injured:** The number of injured related to the event (list of (count,time) values).

We represent our attribute values as timestamped vectors to allow for attributes whose values may be dynamic (e.g. location of a hurricane). **In TREC 2013, the only dynamic target will be location.**

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, as well as the empty lists of attribute values to be tracked (Figure 2).

## 2 Tasks

### 2.1 Temporal Summarization

#### 2.1.1 Task Definition

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

```

<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>
  <location>
    <value>
      <latitude>-34.608639</latitude>
      <longitude>-58.4085</longitude>
      <time>1329910380</time>
    </value>
  </location>
  <dead>
    <value>
      <count>51</count>
      <time>1329910380</time>
    </value>
  </dead>
  <injured>
    <value>
      <count>703</count>
      <time>1329910380</time>
    </value>
  </injured>
</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>
  <location/>
  <dead/>
  <injured/>
</event>

```

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

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 temporal 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), indexing each document in sequence (line 5). If the document we are indexing 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 temporal 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.INDEX(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).

### 2.1.2 Result Format

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

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

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

## 2.2 Value Tracking

### 2.2.1 Task Definition

During the simulation, a system should emit accurate attribute value estimates for an event. Conceptually, a simulator should be structured as in Figure 4. The arguments to the simulator are the participant tracking system, the time-ordered corpus, the keyword query, the

```

TEMPORALSUMMARIZATION(S, C, q, ts, te)
  S           ▷ Participant system.
  C           ▷ Time-ordered corpus.
  q           ▷ Event keyword query.
  ts         ▷ Event start time.
  te         ▷ Event end time.
1  U ← {}
2  S.INITIALIZE(q)
3  for d ∈ C
4      do
5          S.INDEX(d)
6          t ← d.TIME()
7          if t ∈ [ts, te]
8              then
9                  Ut ← S.DECIDE()
10                 for u ∈ Ut
11                     do
12                         U.APPEND(u, t)
13 return U

```

Figure 3: Temporal summarization simulator.

```

VALUETRACKING( $\mathbf{T}, \mathcal{C}, q, a, t_s, t_e$ )
   $\mathbf{T}$            $\triangleright$  Participant system.
   $\mathcal{C}$          $\triangleright$  Time-ordered corpus.
   $q$            $\triangleright$  Event keyword query.
   $a$            $\triangleright$  Event attribute.
   $t_s$          $\triangleright$  Event start time.
   $t_e$          $\triangleright$  Event end time.
1   $\mathcal{V} \leftarrow \{\}$ 
2   $\mathbf{T}.\text{INITIALIZE}(q, a)$ 
3   $v \leftarrow \mathbf{T}.\text{INITIALESTIMATE}()$ 
4   $\mathcal{V}.\text{APPEND}(\langle v, \emptyset \rangle, t_s)$ 
5  for  $d \in \mathcal{C}$ 
6      do
7           $\mathbf{T}.\text{INDEX}(d)$ 
8           $t \leftarrow d.\text{TIME}()$ 
9          if  $t \in [t_s, t_e]$ 
10             then
11                  $\langle v, s \rangle \leftarrow \mathbf{T}.\text{ESTIMATEVALUE}()$ 
12                 if  $\langle v, s \rangle \neq \emptyset$ 
13                     then
14                          $\mathcal{V}.\text{APPEND}(\langle v, s \rangle, t)$ 
15 return  $\mathcal{V}$ 

```

Figure 4: Value tracking simulator.

attribute of interest, and the relevant time range. In line 1, we initialize the value summary to empty. In line 2, we initialize the tracking system with the event query and attribute name. The system should store some representation of this information for later processing and filtering. We then ask for an initial estimate based solely on the query and attribute. Systems are welcome to use any data that existed at or before  $t_s$ . We iterate over the corpus in temporal order (line 5), indexing each document in sequence (line 7). If the document we are indexing is in the event timeframe (line 9), then we check to see if adding the document resulted in a change of the system’s value estimate (line 11). If there is no change in the value estimate, the system should return  $\emptyset$  and continue indexing documents. If the estimate changes, the system should return the new value as well as the id of the supporting sentence.

As with temporal summarization, participants are welcome to implement the experimental framework however they see fit as long as it is well-documented and does not give the tracker access to information unavailable at the time of calling  $\mathbf{T}.\text{ESTIMATEVALUE}()$ .

## 2.2.2 Result Format

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

```
1 HelloWorldUniversity Cluster1 NULL NULL 2013-01-01-14 dead 0
1 HelloWorldUniversity Cluster1 1357052200-54f6f6e096a4cae27bee55dc2e0dc2b6 0 1330432283 dead 2
1 HelloWorldUniversity Cluster1 1357052190-8f7a8b30a9a8f671dcc979677b04fab4 1 1330432283 dead 3
```

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. attribute type
8. attribute value: a floating point number for attributes ‘dead’ and ‘injured’. a comma-separated latitude-longitude value for attribute ‘location’.