HLTCOE at TREC 2013: Temporal Summarization

Tan Xu
University of Maryland
College Park

Paul McNamee
Johns Hopkins University
HLTCOE

Douglas W. Oard
University of Maryland
College Park

Abstract

Our team submitted runs for the first running of the TREC Temporal Summarization track. We focused on the Sequential Update Summarization task. This task involves simulating processing a temporally ordered stream of over 1 billion documents to identify sentences that are relevant to a specific breaking news stories which contain new and important content. In this paper, we describe our approach and evaluation results.

1 Introduction

Temporal Summarization is a new track for this year's TREC evaluation. Its intention is to show a user "what just happened" about a topic in real-time from an evolving data stream. Given a time series set of documents, there are two tasks defined in this track: (1) sequential update summarization, where the goal is to identify sentences that are relevant, novel, and important to an topic of interest; and (2) value tracking, where the goal is to track and emit accurate values for particular attributes of an topic of interest. For this year, we focused only on temporal update summarization. This paper describes our approach and evaluation results in detail.

Update summarization has been a focus of recent automatic summarization research. For example, DUC, and later TAC, included an Update Summarization track from 2007 to 2011 (Dang and Owczarzak, 2008). The task in that track was to generate summaries from a set of newswire articles under the assumption that a user has already read a set of earlier articles. Although the motivation for that track was similar to that of this year's TREC Temporal Summarization task, which is to inform readers of important novel information about a particular topic, the DUC and TAC Update Summarization tasks were designed as a single-pass batch process, processing all new documents at once, while in this year's TREC Temporal Summarization track the task design requires generation of continuous and immediate updates. As with earlier work, sentences are the unit of selection.

Boiling this problem down to its essence, there are three key challenges that any system must address: (1) topicality: select sentences that are about the given topic; (2) novelty: select sentences that contain novel content; and (3) importance: select sentences that a person would put into a summary. In order to address this problem, we designate a set of representative features to capture a sentence's topicality, novelty, and salience, and a composite function $\mathcal{F}$ to synthesize these features into a single-valued decision basis. We then employ a threshold-based approach, which determines whether a sentence should be included in the temporal summary. Both the feature weights and threshold are manually tuned based on the single training topic that was provided to task participants. We extend this basic approach using a number of additional steps to improve effectiveness or efficiency (e.g., Wikipedia-based query expansion, and a preprocessing step designed to efficiently prune non-relevant documents).

2 Approach

Our system is designed by following instruction in the track guidelines,[1] which is structured as in Algorithm 1. The inputs to our system include: a system

[1] http://www.trec-ts.org/

configuration S, the time-ordered corpus $\mathcal{C}$, the topic q, and the time-interval of interest $[t_{start}, t_{end}]$. In line 1, an empty output summary $\mathcal{U}$ is initialized; in line 2, we initialize our system with the topic query. We store a representation of this query for later processing and filtering; in line 3, we iterate over the corpus in temporal order, processing each document in sequence in line 4. If a document is within the specified time-interval (line 5), then we check this document's topicality in line 6. For each document that our system decides is on-topic, an instantaneous decision is made for each sentence of that document about whether to include it in the summary; if so, we note the decision time (line 7-8). Finally, we add the selected sentences to the summary with the time of the decision, and we update our knowledge about the topic (lines 9-11). Below we give more details about the main components of our system.

Algorithm 1: Sequential Update Summarization

```
$\mathcal{U} \leftarrow \{\}$
S.INITIALIZE$(q)$
for $d \in \mathcal{C}$ do
    S.PROCESS$(d)$
    if $d$.TIME() $\in [t_{start}, t_{end}]$ then
        if S.FILTER$(d, q) == true$ then
            for $u \in d$ do
                $u_t \leftarrow$ S.DECIDE$(u)$
                if $u_t == true$ then
                    $\mathcal{U}$.APPEND$(u, t)$
                    S.UPDATE$(q)$
return $\mathcal{U}$
```

2.1 Preprocessing

In 2013, the Temporal Summarization track uses the same document collection as the TREC Knowledge Base Acceleration (KBA) track.[2] This collection contains over a time-series of over 1 billion documents that were obtained from the Web between October 2011 and January 2013 (11,948 hours). Each document in the collection is marked with its access time, which generally was as close as possible to its creation time. Documents that are believed to be written in English have been segmented into sentences and annotated for named entities using the Stanford tagger. The document counts per hour for each composite "source" type is shown in Figure 1. The major types of document in the corpus are newswire articles, social media data aggregated from blogs and forums, and linking records from Bitly.com.

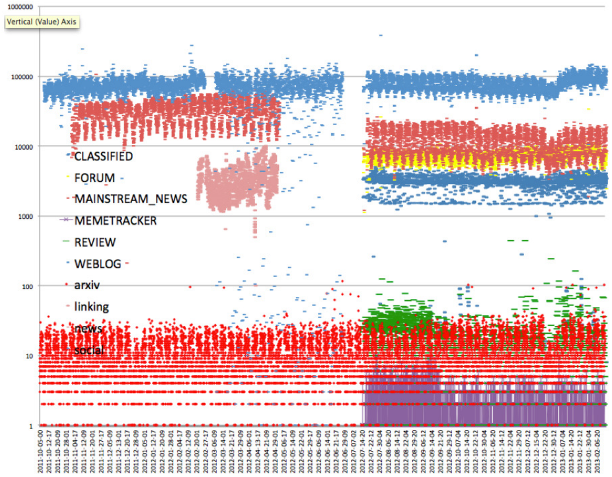


Figure 1: Hourly document counts for TREC KBA Stream Corpus 2013 sources (Frank et al., 2013).

We built a KBA corpus reader to simulate a time-ordered document stream. The original corpus is organized in a shallow hour-structured directory, and within each hourly folder, documents are stored as JSON objects with certain metadata into "chunk" files. Each file contains around 100–300 JSON documents of the same type, and is serialized with Apache Thrift and compressed with XZ Utils. Our corpus reader was developed based on the *streamcorpus* toolkit provided by the TREC KBA track.[3] We first iterate through folders, and then for each chunk file, after decompression and deserialization, we loop over contained documents, and decode each into a Factorie document object with additional POS (Part-of-Speech) tagging.[4] Finally, we sort these document objects according to their timestamp and sequentially pass them to the rest of the system.

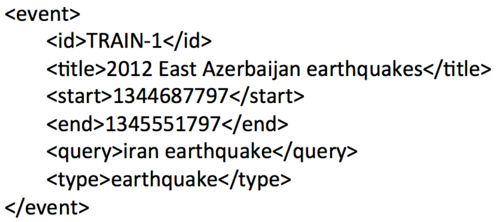
2.2 Topic Representation

In this track, the topics are presented to us in SGML, where the root element is named "event"

[2] http://trec-kba.org/trec-kba-2013.shtml

[3] https://github.com/trec-kba/streamcorpus/

[4] http://factorie.cs.umass.edu/



```
<event>
    <id>TRAIN-1</id>
    <title>2012 East Azerbaijan earthquakes</title>
    <start>1344687797</start>
    <end>1345551797</end>
    <query>iran earthquake</query>
    <type>earthquake</type>
</event>
```

Figure 2: Masked Topic Definition for ‘2012 East Azerbaijan earthquakes’

(because all topics are temporally acute). A topic definition is illustrated in Figure 2, where *title* is a short description of the topic, *query* is a keyword representation of the topic, *type* is one of $\{accident, bombing, earthquake, shooting, storm\}$, and *start* and *end* are the start and ending times for the documents to be processed when building the summary.[5]

We create three Bag-of-Words (BoW) representations for each topic: unigrams (after stopword removal), Named Entities (NE), and predicates (*i.e.,* verbs). Each BoW representation is initialized from the topic’s *title* and *query* fields. As we select sentences for inclusion in the summary, we update each of these BoW representations.

In the topic updating process, one challenge is how best to adapt to the shifting focus of a topic. This problem was also noted in the Topic Detection and Tracking (TDT) evaluations (Allan, 2002). In our work, we tried a basic “Epoch” strategy, as described by Goyal et al., which was initially designed to approximate n-gram frequencies in a streaming setting (Goyal et al., 2009). As a fine-grained implementation of this strategy, we treat the selection of a sentence for inclusion in the summary as an epoch; after each epoch (*i.e.,* each selected sentence), we update each BoW by adding the appropriate terms from the new sentence and then we prune the lowest frequency terms, retaining only the top k terms for each BoW. For our experiments we arbitrarily made the following choices: $k_{unigram} = 1000$, and $k_{NE} = k_{predicate} = 200$.

2.3 Document Filtering

Because of the high rate at which KBA documents were collected (approximately 1,395 documents per minute), we introduce a document filtering stage into our system. We seek to identify irrelevant documents (*i.e.,* those not topically relevant), and preclude any sentences from these documents from further consideration for our temporal summary. To determine a document’s relevance to the topic, we use a cascade of progressively more complex models.

- The first “model” just uses the time-interval specified in the topic to filter out documents that are timestamped before the specified start time or after the specified end time.
- The second model uses Boolean conjunction to filter out documents that do not contain every word in *query* field of the topic.
- The third model calculates the cosine similarity between the unigram BoW vectors for the document and the topic. For each BoW vector, terms are weighted with either TF (term frequency) or TF-IDF (term frequency times inverse document frequency), depending on the system configuration. IDF weights are computed using the Google n-gram corpus (LDC2006T13) (Klein and Nelson, 2008). A threshold is used to determine whether a document should be considered pertinent for the topic.

2.4 Sentence Selection

Sentences should be selected based on three criteria: relevance of the extracted text to the topic, the amount of new information, and the degree to which important aspects of the news event are covered. In order to understand these factors, we manually analyzed the gold standard nuggets selected for the training topic “2012 East Azerbaijan earthquakes” and several Wikipedia pages that report similar types of news (specifically, one of $\{accident, bombing, earthquake, shooting, storm\}$). We examined only Wikipedia pages describing events that predated the KBA collection.

No off-topic or redundant sentences are observed, comporting well with the design of the task, and it seemed to us that named entities and predicates related to the topic might be informative. For example, for the 103 selected nuggets for the training topic, we observed 6 nuggets containing the verb *to kill*

[5] Additional topic fields are available to relevance assessors; the topics provided to systems are referred to as “masked.”

(or one of its inflected forms) and 7 containing some form of *to die*. Both can expected to be a indicative predicate for stories about earthquakes. We also observed that the chance a sentence would be selected was higher if it contained numeric values.

Therefore, although it is still a far-reaching and open-ended question to select an optimal feature set for sentence selection, in this work we focus on a baseline implementation which includes the following features:

- f_1: context document's relevance to the topic, as measured by cosine similarity between the unigram BoW term vectors for the sentence and the dynamically updated unigram BoW term vector for the topic.

- f_2: a sentence's relevance to the topic, as measured by cosine similarity between the sentence's unigram BoW term vector and the topic's initial, static unigram BoW term vector.

- f_3: a sentence's novelty score with regard to previously selected sentences, calculated as one minus cosine similarity between the sentence's unigram BoW term vector and the topic's updated unigram BoW term vector.

- f_4: a sentence's topical salience, calculated using a weighted dot product of named-entities (*i.e.,* effectively a language model from NEs). For example, given a topic $q = \{Iran(2/5), Ahar(2/5), Varzaqan(1/5)\}$, and a sentence *"Iranian state television reported the quake hit near the towns of Ahar, Heris and Varzaqan"*, then $f_4 = (0 + 2/5 + 0 + 1/5)/4 = 0.15$.

- f_5: similar to f_4, this feature estimates salience for a sentence using predicates, where a predicate's topical salience is calculated by its normalized occurrences within the topic's predicate BoW representation.

- f_6: a binary score $\in \{0, 1\}$ that indicates whether a sentence contains numeric values.

We then use convex combination to synthesize the effects of all these features as defined in Equation 1, where λ_i denotes the weight for the ith feature.

$$\mathcal{F}(u_{t+1}|q,\mathcal{U}_t) = \sum_i \lambda_i f_i, \; \| \boldsymbol{\lambda} \|_1 = 1 \tag{1}$$

Because we lacked adequate training data in this first year of the task, we manually tuned $\boldsymbol{\lambda}$ by reviewing system output (*i.e.,* the sentences selected for the summary) for the single available training topic. Figure 3 shows the sentences selected for the first 24 hours of the training topic after hand-optimization of these weights.

2.5 Wikipedia-Based Predicate Expansion

One factor limiting the effectiveness of our basic approach is that the topics are terse, and thus the resulting BoW representations are quite impoverished. Since the gold standard updates are generated based on the revision history of the corresponding Wikipedia page, we imagine that Wikipedia pages for similar events might be a useful source of topic-related vocabulary for our predicate BoW representation. For example, if the topic is about a specific earthquake, we might find that similar words were used to describe important nuggets for previous earthquakes. Therefore, we added a Wikipedia retrieval component to find a small set of topically relevant Wikipedia pages to expand the initial topic. Apache Lucene standard indexing and searching[6] was utilized for this purpose. To avoid using "future" data, this search was based on a Wikipedia dump from October 11th, 2010 (that precedes the KBA Stream Corpus). For each topic, we chose the 10 most highly ranked Wikipedia pages, and extracted predicates to expand query topics.

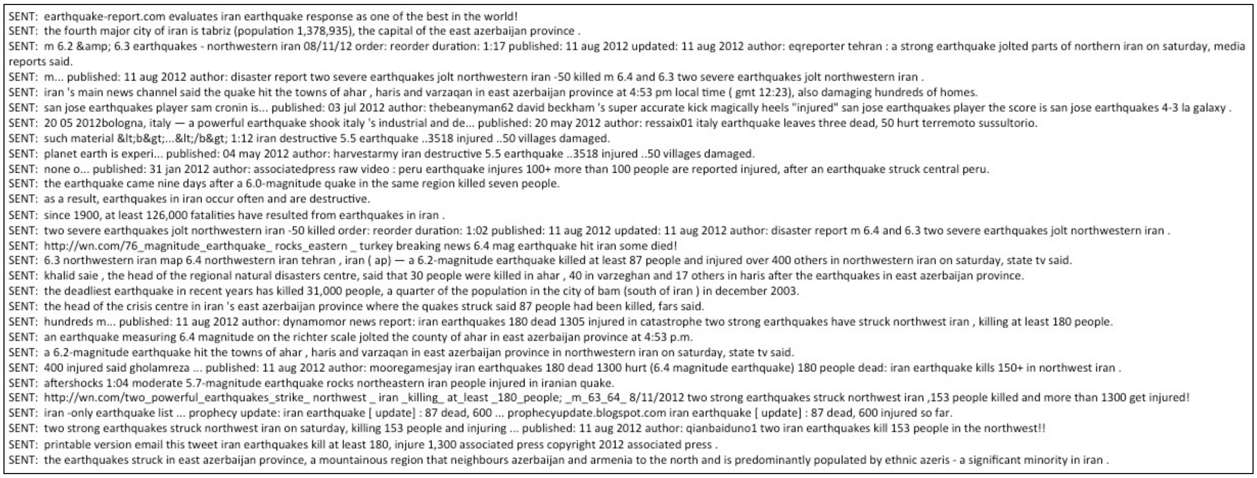
3 Evaluation

We submitted five runs, which are described in section 3.1. In section 3.2, we introduce the track's evaluation metrics for measuring effectiveness. We compare our results to the mean and maximum results provided by NIST in section 3.3.

3.1 Data Set and Submissions

This year's task included 10 topics (2 accidents, 2 shootings, 4 storms, 1 earthquake, and 1 bombing). For each topic, the summarization time window was

[6] http://lucene.apache.org/



```
SENT:  earthquake-report.com evaluates iran earthquake response as one of the best in the world!
SENT:  the fourth major city of iran is tabriz (population 1,378,935), the capital of the east azerbaijan province .
SENT:  m 6.2 &amp; 6.3 earthquakes - northwestern iran 08/11/12 order: reorder duration: 1:17 published: 11 aug 2012 updated: 11 aug 2012 author: eqreporter tehran : a strong earthquake jolted parts of northern iran on saturday, media
reports said.
SENT:  m... published: 11 aug 2012 author: disaster report two severe earthquakes jolt northwestern iran -50 killed m 6.4 and 6.3 two severe earthquakes jolt northwestern iran .
SENT:  iran 's main news channel said the quake hit the towns of ahar , haris and varzaqan in east azerbaijan province at 4:53 pm local time ( gmt 12:23), also damaging hundreds of homes.
SENT:  san jose earthquakes player sam cronin is... published: 03 jul 2012 author: thebeanyman62 david beckham 's super accurate kick magically heels "injured" san jose earthquakes player the score is san jose earthquakes 4-3 la galaxy .
SENT:  20 05 2012bologna, italy — a powerful earthquake shook italy 's industrial and de... published: 20 may 2012 author: ressaix01 italy earthquake leaves three dead, 50 hurt terremoto sussultorio.
SENT:  such material &lt;b&gt;...&lt;/b&gt; 1:12 iran destructive 5.5 earthquake ..3518 injured ..50 villages damaged.
SENT:  planet earth is experi... published: 04 may 2012 author: harvestarmy iran destructive 5.5 earthquake ..3518 injured ..50 villages damaged.
SENT:  none o... published: 31 jan 2012 author: associatedpress raw video : peru earthquake injures 100+ more than 100 people are reported injured, after an earthquake struck central peru.
SENT:  the earthquake came nine days after a 6.0-magnitude quake in the same region killed seven people.
SENT:  as a result, earthquakes in iran occur often and are destructive.
SENT:  since 1900, at least 126,000 fatalities have resulted from earthquakes in iran .
SENT:  two severe earthquakes jolt northwestern iran -50 killed order: reorder duration: 1:02 published: 11 aug 2012 updated: 11 aug 2012 author: disaster report m 6.4 and 6.3 two severe earthquakes jolt northwestern iran .
SENT:  http://wn.com/76_magnitude_earthquake_ rocks_eastern _ turkey breaking news 6.4 mag earthquake hit iran some died!
SENT:  6.3 northwestern iran map 6.4 northwestern iran tehran , iran ( ap) — a 6.2-magnitude earthquake killed at least 87 people and injured over 400 others in northwestern iran on saturday, state tv said.
SENT:  khalid saie , the head of the regional natural disasters centre, said that 30 people were killed in ahar , 40 in varzeghan and 17 others in haris after the earthquakes in east azerbaijan province.
SENT:  the deadliest earthquake in recent years has killed 31,000 people, a quarter of the population in the city of bam (south of iran ) in december 2003.
SENT:  the head of the crisis centre in iran 's east azerbaijan province where the quakes struck said 87 people had been killed, fars said.
SENT:  hundreds m... published: 11 aug 2012 author: dynamomor news report: iran earthquakes 180 dead 1305 injured in catastrophe two strong earthquakes have struck northwest iran , killing at least 180 people.
SENT:  an earthquake measuring 6.4 magnitude on the richter scale jolted the county of ahar in east azerbaijan province at 4:53 p.m.
SENT:  a 6.2-magnitude earthquake hit the towns of ahar , haris and varzaqan in east azerbaijan province in northwestern iran on saturday, state tv said.
SENT:  400 injured said gholamreza ... published: 11 aug 2012 author: mooregamesjay iran earthquakes 180 dead 1300 hurt (6.4 magnitude earthquake) 180 people dead: iran earthquake kills 150+ in northwest iran .
SENT:  aftershocks 1:04 moderate 5.7-magnitude earthquake rocks northeastern iran people injured in iranian quake.
SENT:  http://wn.com/two_powerful_earthquakes_strike_ northwest _ iran _killing_ at_least _180_people; _m_63_64_ 8/11/2012 two strong earthquakes struck northwest iran ,153 people killed and more than 1300 get injured!
SENT:  iran -only earthquake list ... prophecy update: iran earthquake [ update] : 87 dead, 600 ... prophecyupdate.blogspot.com iran earthquake [ update] : 87 dead, 600 injured so far.
SENT:  two strong earthquakes struck northwest iran on saturday, killing 153 people and injuring ... published: 11 aug 2012 author: qianbaiduno1 two iran earthquakes kill 153 people in the northwest!!
SENT:  printable version email this tweet iran earthquakes kill at least 180, injure 1,300 associated press copyright 2012 associated press .
SENT:  the earthquakes struck in east azerbaijan province, a mountainous region that neighbours azerbaijan and armenia to the north and is predominantly populated by ethnic azeris - a significant minority in iran .
```

Figure 3: Summary for '2012 East Azerbaijan earthquakes' (first 24 hours)

limited to 10 days. Our team contributed 5 of the 26 submissions to the track. Shortly after submission, we found that in three of our runs {Baseline, BasePred, EXTERNAL} we had mistakenly calculated f_4 and f_5 by neglecting to normalize the frequency with which named entities or predicates (respectively) were observed in the topic. Because other parameters were set appropriately, those three runs are still useful as a basis for comparison with our other two runs that were normalized correctly {TuneBasePred2, TuneExternal2}. The configurations of each of the 5 runs is given in Table 1.

We experienced one other notable difficulty while producing our runs. In some cases, processing for a topic was prematurely halted due to a memory leak caused by too many in-memory document objects while simulating the temporal stream of documents. The effect of this early termination was to reduce recall somewhat. In every case, the unprocessed documents were those latest in the time window. For sudden-onset events of the type used as topics this year, the reporting is often concentrated early in the period. As a result, the adverse effect on recall of our unintended early termination (when it occurred) might be far less than the loss of temporal coverage might otherwise suggest. Table 2 reports the fraction of the time window that was actually processed for each topic in each submitted run.

3.2 Evaluation Metrics

Traditional evaluation measures for automatic summarization such as ROUGE (Lin, 2004) focus on the presence or absence of a sentence in a summary. In the Sequential Update Summarization task, by contrast, the key question is about latency (with absence simply being an extreme case of latency). A set of gold standard updates (nuggets) were manually extracted from the Wikipedia page corresponding to the event that is the focus of the topic. Each update is timestamped according to the revision history of that page. A generated sequential-update summary is a set of sentences, each timestamped by the decision time. The evaluation measures are thus analogous to the traditional set-based measures of precision and recall, but extended to include a latency penalty.

More specifically, following the track guidelines, we evaluate effectiveness using Expected Latency Gain (EG_L), which is similar to traditional notion of precision, and Latency Comprehensiveness (C_L), which is similar to traditional notion of recall, between the summaries produced by human annotators (N) and our system (S).

$$EG_L(S) = \frac{1}{|S|} \sum_{\{n \in N : M(n,S) \neq \phi\}} g_L(M(n,S), n) \qquad (2)$$

$$C_L(S) = \frac{1}{\sum_{n \in N} R(n)} \sum_{\{n \in N : M(n,S) \neq \phi\}} g_L(M(n,S), n) \qquad (3)$$

Table 1: Parameter settings for each run

		External Resource		Feature Weights						Sentence
	Predicate	IDF	Wikipedia	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	Threshold
TuneBasePred2	✓			0.24	0.28	0.12	0.13	0.20	0.03	0.30
TuneExternal2	✓	✓	✓	0.24	0.28	0.12	0.13	0.20	0.03	0.30
Baseline				0.30	N/A	0.30	0.30	N/A	0.10	0.20
BasePred	✓			0.23	N/A	0.23	0.23	0.23	0.08	0.20
EXTERNAL	✓	✓	✓	0.23	N/A	0.23	0.23	0.23	0.08	0.20

Table 2: Fraction of documents processed for each topic, by run

	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10
TuneBasePred2	100%	87%	38%	11%	100%	100%	100%	100%	100%	100%
TuneExternal2	100%	100%	77%	10%	100%	82%	100%	100%	100%	100%
Baseline	100%	23%	9%	11%	91%	63%	100%	100%	100%	100%
BasePred	100%	23%	9%	11%	94%	63%	100%	100%	100%	100%
EXTERNAL	100%	24%	27%	10%	53%	61%	100%	94%	100%	78%

$M(n, S)$ denotes the earliest matching update u from our system to a given gold standard nugget n, which can be expressed as $argmin_{\{u \in S : n \approx u\}} u.t$. $g_L(u, n)$ denotes latency-discounted gain getting from u for n, computed as $(u.t - n.t) \times R(n)$, where $R(n)$ denotes the importance of n. In N, each nugget has an associated relevance grade assigned by human annotators, $R : N \to [0, 1]$.

3.3 Results

The results for our five submissions are plotted in Figure 4, where for each evaluation topic $q1 \sim q10$, the solid triangle, circle and square points represent the NIST reported maximum, average and minimum EG_L and C_L scores over all TREC submissions respectively. [7] The curved lines show contours at intervals of 0.1 points of the balanced harmonic mean of the two measures.[8] We omit topic 7, which all participants did poorly on because there were not enough (detected) relevant documents within the specified time window.

As Figure 4 shows, we generally did well on topics 3 and 10 by the EG_L (precision-like) measure and on topics 1 and 10 by the C_L (recall-like) measure; we did poorly on topic 5 by both measures. Interesting, the three runs in which we mistakenly failed to normalize ($\Box$ BasePred, $\circ$ Baseline, $\triangle$ EXTERNAL) yielded relatively high C_L scores. The lower C_L scores for our other two runs ($+$ TuneBasePred2, $\times$ TuneExternal2) can not be explained by early termination, since the other three unintentionally unnormalized runs have similar (or more severe) early termination. As Table 1 shows, the threshold we selected (after examining sample output) was higher for the two “properly” normalized runs. From this we can infer that our “properly” normalized runs are more conservative about allowing sentences into the summaries, although we do not at this point know whether that is because we are computing scores differently or that we set the threshold to different values. We should also note that our manual parameter selection was based on getting results that “looked good” to us, and of course we would be more likely to notice bad selections than to notice what was missing. As a result, we may have been precision-biased in our parameter selections. The fact that our two “properly” normalized runs do better by the EG_L measure comports with that speculation. We note similar effects from the use of IDF and Wikipedia query expansion regardless of whether correct normalization was applied (see Table 1 for run configurations).

Focusing now on the two “properly” normalized runs, and especially for topics $q1, q8, q9$ and $q10$, which did not suffer from early termination, another observation is that the use of IDF increased EG_L

[7] Note: The MAX and MIN values reported by NIST are computed over each measure independently. Because both recall-tuned and precision-tuned systems contributed runs, plotting the MAX values for both measures as a single point is not indicative of what any single system achieved.

[8] If these were precision and recall, these would be F_1 contours; they are calculated by $2 \cdot EG_L C_L / (EG_L + C_L)$).

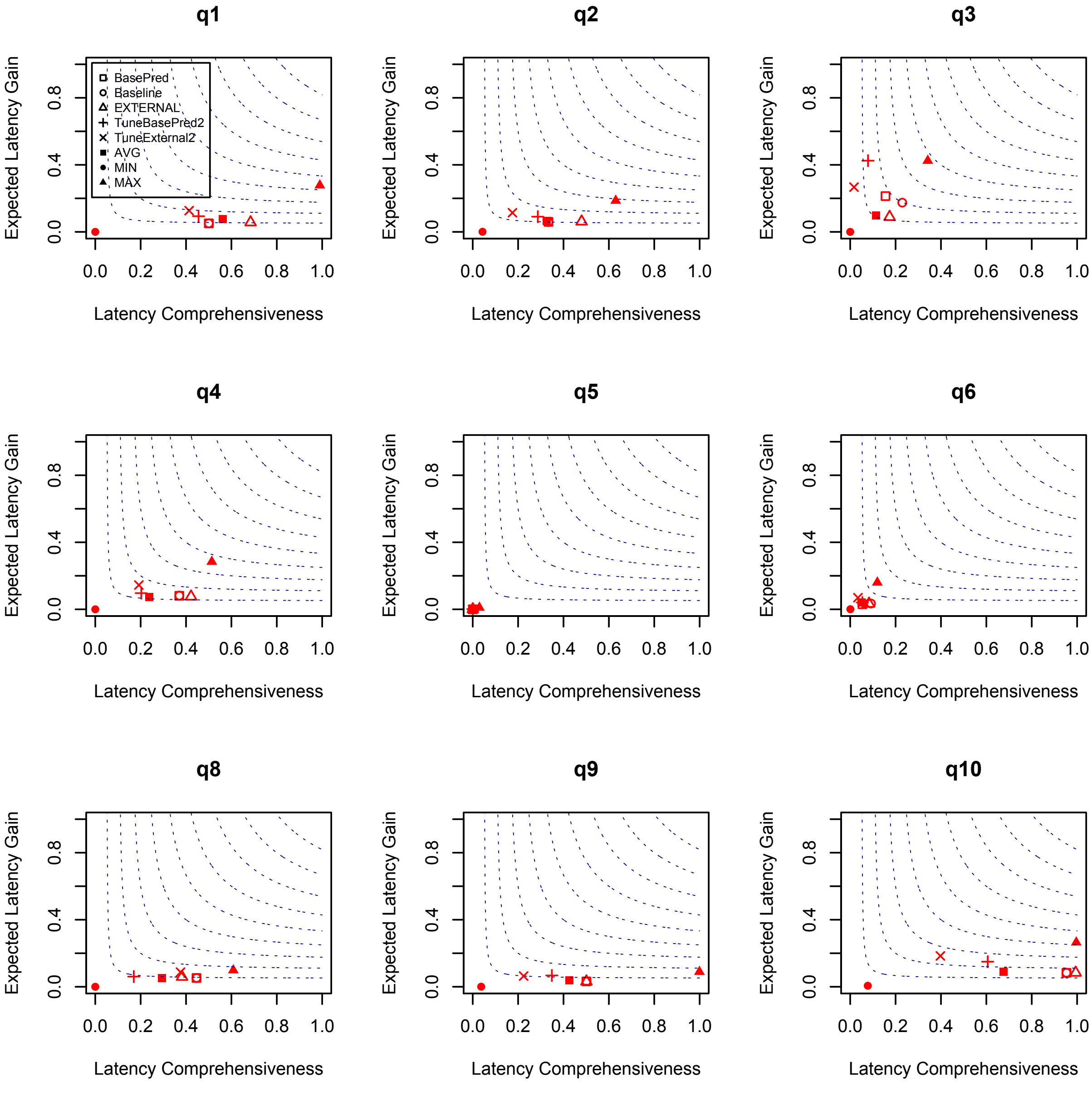


Figure 4: Sequential update summarization evaluation results, EG_L and C_L scores.

(the precision-like measure). However, Wikipedia-based predicate expansion did not increase the C_L score as we had expected it would. Indeed, predicate expansion decreased C_L in most cases (the exception being q8). Inspection of the retrieved Wikipedia pages that were used to expand these query topics revealed that the top 10 returned pages were often about similar entities rather than similar events. Thus the predicates extracted from these pages did not provide event-focused information as we had hoped, but rather added noise. We believe that idea still has merit, but our technique needs refinement.

Looking more broadly at our approach, our sentence selection model can be thought of as a variant of Maximal Marginal Relevance (MMR), where the key idea is to balance relevance and novelty (Carbonell and Goldstein, 1998); in our case, we must also balance salience. Similar to MMR, we measure a sentence's novelty by considering its difference from past sentences (represented by updated unigram BoW, as described in section 2.2). However, as these past sentences were themselves selected according to their topicality, relevance, and novelty, they are inextricably linked by the nature of the evidence that we use. This issue has also been observed by Allan et al. in their early work of temporal summarization (Allan et al., 2001).

4 Conclusions

For this first running of the Temporal Summarization track at TREC, we designed an extractive summarization system using a simple linear model and straightforward features to detect sentences that contain novel and salient information. These sentences come from a large streaming collection, and our system makes binary decisions about each incoming document in real-time as it arrives. We explored dynamic updating of the topic representation as sentences were selected, and we tried a variant of query expansion using Wikipedia pages. The scale of the data posed some challenges, but we have been able to draw some useful insights from our results. Our analysis of those results to date suggests several areas for future work, including: (1) optimizing both document and sentence selection thresholds; (2) finding better exemplars for similar (historical) events in Wikipedia (e.g., by exploiting the Wikipedia category system); (3) designing additional features to represent a sentence's properties of topicality, novelty, and topical salience; and (4) investigating more sophisticated models for sentence extraction. With the new labeled data from this year's track, our work is just beginning.

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Topic models for summarizing novelty. In *ARDA Workshop on Language Modeling and Information Retrieval*, Pittsburgh, PA, USA.

James Allan. 2002. Introduction to topic detection and tracking. In James Allan, editor, *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 1–16. Springer.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of Text Analysis Conference*, TAC 2008, pages 1–16, Gaithersburg, MD, USA. NIST.

John R. Frank, Steven J. Bauer, Max KleimanAWeine, Daniel A. Roberts, Nilesh Tripuraneni, Ce Zhang, Christopher Re, Ellen M. Voorhees, and Ian Soboroff. 2013. Evaluating stream filtering for entity profile updates for TREC 2013. In *TREC The Text Retrieval Conference*, Gaithersburg, MD, USA. NIST.

Amit Goyal, Hal Daumé, III, and Suresh Venkatasubramanian. 2009. Streaming for large scale nlp: language modeling. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 512–520, Stroudsburg, PA, USA. ACL.

Martin Klein and Michael L. Nelson. 2008. A comparison of techniques for estimating IDF values to generate lexical signatures for the web. In *WIDM '08: Proceeding of the 10th ACM workshop on Web information and data management*, pages 39–46, New York, NY, USA. ACM.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. ACL.