Explicit Diversification of Event Aspects for Temporal Summarization

RICHARD MCCREADIE, University of Glasgow
RODRYGO L. T. SANTOS, Universidade Federal de Minas Gerais
CRAIG MACDONALD, University of Glasgow
IADH OUNIS, University of Glasgow

During major events, such as emergencies and disasters, a large volume of information is reported on newswire and social media platforms. Temporal summarization (TS) approaches are used to automatically produce concise overviews of such events, by extracting text snippets from related articles over time. Current TS approaches rely on a combination of event relevance and textual novelty for snippet selection. However, for events that span multiple days, textual novelty is often a poor criterion for selecting snippets, since many snippets are textually unique, but are semantically redundant or non-informative. In this article, we propose a framework for the diversification of snippets using explicit event aspects, building upon recent works in search result diversification. In particular, we first propose two techniques to identify explicit aspects that a user might want to see covered in a summary for different types of event. We then extend a state-of-the-art explicit diversification framework to maximize the coverage of these aspects when selecting summary snippets for unseen events. Through experimentation over the TREC TS 2013, 2014 and 2015 datasets, we show that explicit diversification for temporal summarization significantly outperforms classical novelty-based diversification, as the use of explicit event aspects reduces the amount of redundant and off-topic snippets returned, while also increasing summary timeliness.

CCS Concepts: •**Information systems** → **Summarization;**

Additional Key Words and Phrases: Temporal Summarization, Explicit Diversification, xQuAD

ACM Reference format:
Richard McCreadie, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2017. Explicit Diversification of Event Aspects for Temporal Summarization. *ACM Transactions on Information Systems* 36, 3, Article 25 (November 2017), 33 pages.
DOI: 10.1145/3158671

1 INTRODUCTION

During large unforeseen events, such as natural disasters, there is extensive reporting in both classical newswire outlets and social media. However, far more content is published than can be consumed by a user following such an event. Moreover, much of the information reported is redundant and/or out-of-date [18]. Hence, there is a need for automatic approaches that summarize that content for the user in real-time.

However, in practice, effectively summarizing large events using content published online is a challenging problem. First, in contrast to the multi-document summarization (MDS) domain [14],

where it is assumed that all of the input content is relevant, much of the content collected (e.g. via keyword tracking on social media platforms such as Twitter) can be off-topic. This renders classical MDS approaches ineffective, since they primarily use text novelty to select content for inclusion into the summary (the content that is the most textually novel will likely be off-topic). Second, an effective summarization system needs to identify different types of relevant information over time. For example, consider a user following the mass shooting that occurred in Aurora, Colorado in July 2012.[1] At the outset of this event, an effective summary of the event should contain information such as 'At least 12 people were shot in the city of Aurora near Denver, Colorado'. However, a few hours later, the user might want to know that 'One suspect wearing a bulletproof vest was apprehended' or 'Once apprehended, the suspect told authorities that there were explosives in his residence'. Hence, an effective summarization system needs to automatically identify content that is both relevant and covers the different types of information that the user is likely to be interested in seeing for that particular event.

In this article, we propose a new temporal summmarization framework that automatically extracts sentences from news and social content published on the Web over time to issue as updates to the user. In particular, building on recent works in the field of explicit Web search diversification [36], we propose to select sentences based on their coverage and novelty with respect to a predefined set of *event aspects*, tailored to the type of the event being summarized. For example, for a mass shooting event (like the Aurora shooting discussed above), based on prior shootings in the U.S., we can predefine a taxonomy describing the types of information that a user would want to know about, such as how many people were killed or injured, who the shooter was and how the government responded. Based on these aspects, given a candidate sentence, we score it based on how relevant to the event it is predicted to be, how many aspects of that event it covers and how many of those aspects have been covered by previous updates returned to the user.

The primary contributions of this article are four-fold:

(1) We propose explicit diversification as a technique for performing event summarization over time
(2) We propose two different methods to generate explicit event aspects for named event types
(3) We extend a state-of-the-art diversification framework, namely xQuAD [35] for the task of sentence selection during summarization
(4) We experimentally evaluate the performance of this approach over 46 events from the TREC 2013-2015 Temporal Summarization track datasets, showing that the use of explicit event aspects reduces redundant and off-topic content in the resulting summaries, while also increasing summary timeliness.

The remainder of this article is structured as follows. Section 2 discusses related works in the field of summarization and diversification. In Section 3, we describe a basic temporal summarization framework that forms the baseline approach that we build on. Section 4 details our proposed explicit diversification framework for temporal summarization. In Section 5, we describe our experimental setup, while Section 6 details our label generation methodology. We report and analyze the performance of our approach in Section 7. Finally, we summarize our main conclusions in Section 8.

2 RELATED WORK

The approach that we propose in this article builds on top of prior works in both the summarization and Web search diversification domains. We first provide an overview of automatic summarization

[1] http://en.wikipedia.org/wiki/2012_Aurora_shooting

techniques. In particular, to aid structuring, we categorize past topic/event summarization works into four main groups, based on the algorithmic assumptions made, namely: Multi-Document Summarization works, Update Summarization works, Timeline Generation works and Temporal Summarization works. The commonality between all of these works is that they attempt to summarize a topic or event based on textual documents.[2] We then discuss Web search diversification techniques and how we build upon them for the temporal summarization task investigated here.

2.1 Multi-Document Summarization

Early literature examining summarization primarily focused on multi-document summarization (MDS), which takes as input a set of (clean) documents about a topic to be summarized and generates a fixed length summary, normally by extracting sentences from those documents [8, 11, 27, 31, 43, 46]. The MDS task was originally proposed as a task at the Document Understanding Conference (DUC)[3] and then further investigated at the Text Analysis Conference (TAC) [14]. Popular approaches to MDS typically involve the calculation of a centroid vector from the input documents, representing the main topic of the event. A sentence is then selected based on how similar it is to the centroid [26, 34], i.e. how topical that sentence is. Other approaches identify statistically important terms based on the corpus statistics using techniques such as Probabilistic Latent Semantic Analysis [9]. Machine-learned sentence selection approaches that combine multiple text analysis features such as entity presence extracted from the sentences themselves have also been examined and shown to be effective [24, 38, 45].

2.2 Update Summarization

One of the short-comings of MDS approaches is that they are retrospective in nature, i.e. they assume that all of the relevant content is available beforehand as input, when in practice many applications call for summaries to be updated over time as new information emerges [2]. To address this, the update summarization task was proposed, which aims to produce additional fixed-length 'update' summaries from documents published later in time, covering new information [14]. As with MDS techniques, these update summarization assumes that all of the input documents are on-topic. Update summarization approaches [14, 24, 42, 49] generally start by applying an MDS approach over the new documents, followed by a redundancy removal technique. The most common method for removing redundancy is to apply a variant of Maximal Marginal Relevance (MMR) [7]. MMR incrementally selects sentences in a greedy manner, where, in each iteration, the sentence that is most textually dissimilar (novel) to those previously selected is chosen. Notably, MMR is also used for implicit diversification in IR, as discussed later in Section 2.5. Under MMR, a tuning parameter, λ, controls the trade-off between relevance and novelty. There is also a second class of update summarization approaches that use clustering techniques to create a representation of the key information about the event. For instance, Wang and Li [42] used incremental hierarchical clustering over the sentences within each document. These clustering-based approaches implicitly diversify by only selecting one update from each cluster to include in the summary.

2.3 Timeline Generation and Timeline Summarization

A separate line of research that stems from MDS is timeline generation/summarization. The aim of timeline generation is to produce an itemized timeline for a major event based on a set of documents about that event. While the intent differs slightly from MDS, the task is functionally

[2] Note that there is a separate line of research that attempts to summarize all events that are occurring during a time period, such as [19], rather than summarizing a single event or topic that we examine here.
[3] http://www-nlpir.nist.gov/projects/duc/

identical, and MDS approaches can be directly applied. New approaches developed for timeline generation combine a wider range of factors to select sentences. For instance, the evolutionary timeline generation approach proposed by Yan et al. [49] combines relevance, coverage, coherence and diversity (with respect to the previously selected sentences) features within a machine learning framework. Meanwhile, Tran et al. [39] examined how to generate timelines by extracting headlines from news articles based on a variant of Topic-Sensitive PageRank [17] designed to capture whether each headline is influential and widely spread. Other works have examined the generation of timelines for social media data such as tweets using topic clustering and social features [23]. The closest prior work from this domain to the approach proposed in this article is the aspect-oriented approach proposed by Li et al. [25]. They proposed the use of LDA topics (that they refer to as event aspects) to cluster sentences, using those clusters as a means to avoid selecting redundant content. In contrast, the approach proposed in this article seeks to diversify the selected sentences in a temporal summarization setting with respect to explicitly pre-identified event aspects, such as the number of people killed, building damage sustained or location, about which the user will likely want to see related information in a summary. The key motivation for this is that rather than relying on approaches like LDA, we can predict the aspects of interest for a future event, by examining past events of that same type.

2.4 Temporal Summarization

In 2013 the Text Retrieval Conference (TREC) introduced the Temporal Summarization (TS) track that examined how to extract sentences from high volume streams of news and social content to return to the user as updates for large events. Unlike the summarization tasks described above, the TS task does not assume that the input document stream is on-topic. Each event is represented by an event query, e.g 'costa concordia disaster'. Temporal summarization can be seen as a sentence scoring problem, where participant systems score each sentence and then emit those with scores above a predefined threshold. For example, Liu et al. [28] used the event query in addition to trigger words such as 'kill', 'die' and 'injure' to score individual sentences, while Xu et al. [48] combined features such as document relevance, sentence relevance and topical salience based on named entities for sentence scoring. Meanwhile, recent work by Zhang et al. [51] examined LDA topic modelling approaches to mine event words for sentence scoring. After the first year of the TS track, supervised approaches to sentence scoring became popular. For instance, McCreadie et al. [30] used a learning-to-rank function combining relevance, quality and novelty features to rank sentences and then used a supervised predictor to determine how many of the top-ranked results to return during different time intervals (e.g. each hour). Meanwhile, Kedzie et al. [21] proposed an approach that learns a sequential decision making algorithm for use over a binary branching tree representing the sentences in the input stream. Regression models for identifying topical and good quality sentences have also been popular [22, 40]. For instance, Kedzie et al. [22] used a Gaussian Process regression model with a combination of query features, event-type-based language model features, geographic features and temporal burstiness features to score sentences. Indeed, we use this model as a baseline in our later experiments.

One element that is common to all of these approaches is that they apply a final novelty filtering step, with the aim of removing redundant content [22, 28, 30, 48], similarly to the multi-document summarization approaches that came before them, e.g. [24, 38]. In contrast to these works, we propose to use explicit diversification of event aspects to find novel content, rather than relying on textual dissimilarity. We argue that explicit diversification of event aspects should be more effective for temporal summarization, since explicit approaches have been shown to be more effective than implicit approaches in the Web search domain [10, 36]. Moreover, in a real-time streaming

setting, where much of the considered content will be off-topic, the most novel-looking content is often not relevant.

2.5 Diversification in IR

Search result diversification aims to increase the chance that a particular search engine result page (SERP) will satisfy the user by including documents that cover different possible query aspects when the user query is ambiguous [36]. For instance, for the query 'jaguar', it may not be clear whether the user is looking for documents about the cat or the car manufacturer, hence the SERP should include documents satisfying both aspects.

Approaches to search result diversification can be divided into two main types, namely implicit or explicit [36]. Implicit diversification approaches attempt to diversify based on a representation of the retrieved documents. Indeed, Maximal Marginal Relevance (MMR) [7] is a well known implicit diversification approach that we discussed earlier. Other implicit diversification approaches have incorporated the expected mean and variance of the SERP, inspired by Modern Portfolio Theory [44, 52].

On the other hand, explicit diversification approaches [1, 35] compare documents to (explicit) representations of the possible aspects behind the query. These aspects are often extracted from query logs [35, 36] or taxonomies like the Open Directory Project (ODP) [1]. Effective explicit diversification approaches from the literature include Explicit Query Aspect Diversification (xQuAD) [35], IA-Select [1] and PM-2 [15].

In this article we propose to use explicit diversification techniques for temporal summarization, with the aim of enhancing coverage of different event aspects in the summary. In particular, later in Section 4.2, we show how the xQuAD [36] and IASelect [1] diversification approaches can be adapted to select sentences for inclusion into a summary. The core novelty of this work is the use of explicit event aspects for temporal summarization, and adaptation of explicit Web search diversification techniques to achieve this. To the best of our knowledge, explicit event aspects have not been used previously for temporal summarization. The closest work to ours is that by Li et al. [25] in the timeline generation domain, who used LDA topics to cluster sentences (creating a form of event aspect) and then return an exemplar from each cluster to form the summary. However, this type of approach is not applicable to the temporal summarization scenario, as much of the input content is off-topic, hence the resulting clusters will not well capture the aspects of the event. Another relevant related work is that by Kedzie et al. [22], who use event-type language models derived from Wikipedia. This holds similarities to the Wikipedia-based event-aspect extraction approach we use later, but these two uses of Wikipiedia are very different. In particular, Kedzie et al. [22] used event-type language models as a way to tackle term mismatch between the event query and the text of each update, thereby making the identification of relevant updates easier. In contrast, we extract explicit aspect representations from Wikipiedia info-boxes, creating a structured representation of key information about the event. We then track the extent to which each of these explicit aspects have been covered by the previously submitted updates, enabling us to promote updates that cover previously unseen or under-represented aspects.

In the next section we define the task formulation that we tackle in this article and describe the basic summarization framework which provides the structure of our investigation into temporal summarization.

3 BASIC TEMPORAL SUMMARIZATION FRAMEWORK

The summarization task that we tackle is temporal summarization, as defined by the TREC Temporal Summarization track. Formally, a temporal summarization system is given as input the following information:

- An event query Q representing an event e that the user wants to track, e.g. Buenos Aires Train Crash.
- The type of the event, denoted e_t, e.g. a storm or earthquake.[4]
- A stream of documents published over time D.

The aim of the temporal summarization system is to extract sentence updates u from D to return to the user, forming a final summary S. The resulting summary should contain as little redundant or off-topic information as possible.

To form a basis for a later comparison, we first define a basic temporal summarization framework that represents approaches to this task from the literature. In particular, the most common type of approach to temporal summarization is a 'rank-then-select' approach [28, 30]. Under this type of approach, the input document stream D is indexed over time. At pre-defined intervals (e.g. at the end of each hour), candidate updates produced during that time interval are subject to a selection criteria *select(u)*. Finally, candidate updates that pass the selection criteria are then subject to standard novelty-based redundancy removal based on greedy cosine similarity comparisons between each update u and those updates previously selected in S [28, 30, 48]. We implement this 'rank-then-select' style approach as our basic temporal summarization framework and use it to generate baseline approaches to compare against in our experiments.

Within the basic temporal summarization framework, the component that we investigate in this article is the selection criteria *select(u)*, which outputs a binary true/false choice of whether to include u in the summary. In general, approaches to calculate *select(u)* can be divided into three parts:

(1) **Scoring Function**: This scores an individual update based on a definition of 'goodness' for inclusion into the summary. Most commonly this is relevance to the event query Q, but more complex learned combinations that incorporate other factors such as writing quality or salience have also been examined [22, 30];
(2) **Selection Criterion**: A method for selecting a subset of the scored updates for inclusion into the summary. The most commonly used are *topk* selection that ranks the updates by their scores and then selects the top k updates, and $threshold$ selection, that selects updates whose score exceeds a pre-defined threshold value τ [30].
(3) **Redundancy Removal**: An additional check that removes updates which contain redundant information to those already provided to the user, typically based on the cosine similarity to sentences previously selected [28, 30, 48].

In our later experiments, we generate baselines that combine either relevance or salience-based scoring functions with *topk* and $threshold$ selection criteria and classical novelty-based redundancy removal. We contrast these baselines with our newly proposed approach that introduces a novel scoring function that incorporates the explicit diversification of event aspects, eliminating the need for a separate redundancy removal component.

[4]The TREC Temporal Summarization track dataset that we use for evaluation provides a natural language label specifying the event type for each query.

4 EXPLICIT DIVERSIFICATION FOR SUMMARIZATION

Inspired by growing literature on explicit search result diversification in the Web domain [1, 35], we propose to score each update based on its coverage and novelty with respect to a set of *event aspects* (denoted $\mathcal{A}$) that represent the different types of information a user might want to see in a summary for the event, rather than relying only on implicit novelty like prior approaches. For example, if we want to summarize an earthquake event, we might want to identify aspects such as the earthquake epicentre, the magnitude of the earthquake or the number of people injured.

To achieve this, we propose a new framework for explicit diversification of event aspects for temporal summarization. This framework is comprised of two main components:

- **Event Aspect Generation**: An approach to automatically identify the types of information (event aspects $\mathcal{A}$) the user might want to see for the event being summarized.
- **Explicit Diversification**: A technique to incorporate these event aspects when scoring each update, i.e. a model for estimating the score for an update u, denoted $f(q, u, \mathcal{S})$, given $\mathcal{A}$.

In Section 4.1, we discuss two ways to tackle the Event Aspect Generation component, while Section 4.2 details how we adapt the xQuAD framework for use within the Explicit Diversification component.

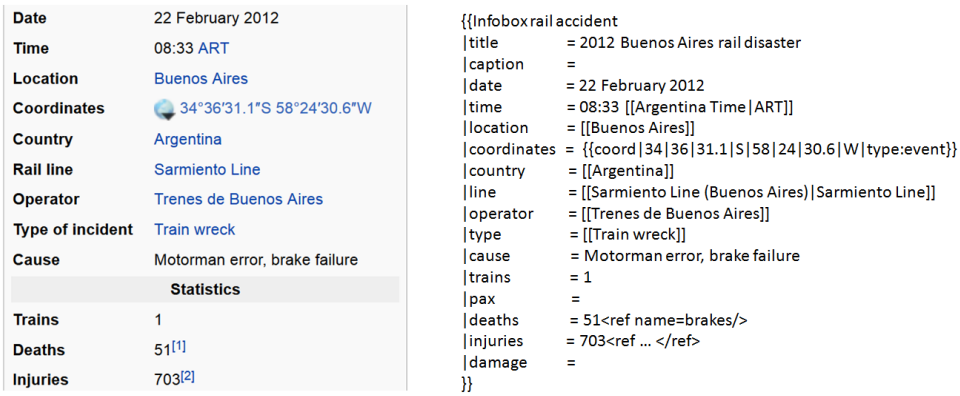
4.1 Event Aspect Generation

The first component of our framework is Event Aspect Generation (EAG). The aim of this component is to generate a set of event aspects ($\mathcal{A}$) for the event that is to be summarized, where these event aspects will be used to diversify the types of information that the user will see in their summary. In the Web search domain, automatically generated query suggestions/reformulations for the initial query have been a popular source of query aspects [10, 32, 36]. However, in the summarization domain, query suggestions are less likely to be useful, since when an unexpected event first occurs the suggestions generated are unlikely to be relevant (as query suggestions are typically based on historical querying behavior by users [6]). Hence, we need an alternative evidence source from which to extract event aspects. Furthermore, as we are working in a real-time (streaming) setting, we cannot 'look into the future' to find out what the actual event aspects that the user might want to see are.

In the temporal summarization setting, we have two pieces of evidence at the start of the event that we can use to find relevant event aspects $\mathcal{A}$, namely the event query Q and the type of the event e_t. In this work, we propose to use the type of the event e_t to generate relevant event aspects by analyzing texts for past events of the same type.[5] For the purposes of our later experiments, following the standard TREC TS setting (and also to limit the number of independent variables), we use the 10 unique event types defined by the TREC TS dataset. However, as an aside, it is of note that in scenarios where an event type label is not provided a substitute might be automatically inferred. For instance, one approach to infer the event type would be to compare language models [12] generated for common event types to the query Q. Supervised classification techniques could also be used to infer the event type [41].

We propose two different methodologies for generating the event aspects $\mathcal{A}$ for a given event type, and compare their effectiveness later in Section 7. We discuss each methodology in detail below.

[5] Note that, although we consider statically mined event aspects in the present investigation, our approach could naturally benefit from mining new aspects (perhaps, event-specific) in real-time. We leave the investigation of such a temporal aspect mining approach for future research.



Date	22 February 2012
Time	08:33 ART
Location	Buenos Aires
Coordinates	34°36′31.1″S 58°24′30.6″W
Country	Argentina
Rail line	Sarmiento Line
Operator	Trenes de Buenos Aires
Type of incident	Train wreck
Cause	Motorman error, brake failure
Statistics	
Trains	1
Deaths	51[1]
Injuries	703[2]

(a)

```
{{Infobox rail accident
|title        = 2012 Buenos Aires rail disaster
|caption      =
|date         = 22 February 2012
|time         = 08:33 [[Argentina Time|ART]]
|location     = [[Buenos Aires]]
|coordinates  = {{coord|34|36|31.1|S|58|24|30.6|W|type:event}}
|country      = [[Argentina]]
|line         = [[Sarmiento Line (Buenos Aires)|Sarmiento Line]]
|operator     = [[Trenes de Buenos Aires]]
|type         = [[Train wreck]]
|cause        = Motorman error, brake failure
|trains       = 1
|pax          =
|deaths       = 51<ref name=brakes/>
|injuries     = 703<ref ... </ref>
|damage       =
}}
```

(b)

Fig. 1. Example Wikipedia Infobox.

4.1.1 Extracting Aspects from Wikipedia Infoboxes. We first experiment with a fully automatic approach for extracting event aspects based upon Wikipedia infoboxes. A Wikipedia infobox is a table structure that appears in some Wikipedia pages, which provides relevant factoids about the page subject. Figure 1 (a) provides an illustration of the Wikipedia infobox for the 2012 Buenos Aires rail disaster page as rendered on Wikipedia. The Wikipedia infobox for a page can be programatically accessed via the Wikipedia API.[6] Figure 1 (b) illustrates the source text of the same Wikipedia infobox for the 2012 Buenos Aires rail disaster page. Importantly, Wikipedia provides templates to use when creating infoboxes, which standardizes the types of information provided. For example, the 'event' infobox template contains fields such as 'Time', 'Cause' and 'Deaths'.[7] As a result, similar types of information are provided in infoboxes for different events.

Wikipedia infoboxes have been popularized as a useful source of factual information [47] and are used by knowledge bases such as DBpedia [5]. Furthermore, most event-related pages contain infoboxes. For example, for the Wikipedia page about the '2012 Buenos Aires rail disaster'[8] that is shown in Figure 1, we can see that it contains useful factoids such as 'Location', 'Rail line', and 'Deaths'. We propose to extract candidate event aspects from the infobox factoids across multiple Wikipedia pages for a target event type.[9]

In particular, we use the event type e_t (e.g. 'earthquake' or 'train crash') as a query over Wikipedia, with the aim of finding Wikipedia pages that describe past relevant events of that type. We then use the factoids from the infoboxes on those pages to produce our event aspects A. Figure 2 illustrates the event aspect extraction process as pseudo code. As we can see from Figure 2, we first identify Wikipedia pages discussing past events of a given type by searching Wikipedia using the event type as a query. For each of the top 10 pages, we check whether it has an infobox. If so, we extract all of the <key,value> pairs representing the factoids listed in

[6] https://www.mediawiki.org/wiki/API:Main_page
[7] https://en.wikipedia.org/wiki/Template:Infobox_event
[8] https://en.wikipedia.org/wiki/2012_Buenos_Aires_rail_disaster
[9] Note that an alternative approach might be to use the infobox from the Wikipedia page for each event directly, i.e. if an event has a page describing that event, then using the infobox from that page to infer the event aspects. However, not all events of interest have Wikipedia pages, particularly early on during those events. Furthermore, prior research has indicated that when such pages exist they can be slow to be updated [33].

Wikipedia Event Aspect Extraction

1: **Input**
    Event type, *type*
2: **Output**
    Event aspects A, a mapping between each aspect and one or more representations, $l \in \mathrm{L}$
3: Search Wikipedia using *type* as the query $\rightarrow$ *Docs*
4: for each *Doc* in Top10(*Docs*) loop
5: if containsInfobox(*Doc*) then
6: for each <*key*,*value*> from parseInfobox(*Doc*)
7: cleanFactoidValues(*value*) $\rightarrow$ *cleanValue*
8: if not A.contains(key) A.put(key, $\varnothing$)
9: A.get(key).add(*cleanValue*)

Fig. 2. Algorithm for extracting Event Aspects from Wikipedia

the infobox. For instance, for the rail disaster example above, one factoid has the key 'Cause' and value 'Motorman error, brake failure'. For each factoid key, we add a new event aspect to A (if an aspect with that name was not already added). We then clean any event specific data from the factoid value that is unlikely to generalize to future events. In particular, we apply a part-of-speech tagger and named entity recognition algorithm over the terms in the factoid value, and then filter out any named entities, numeric values and URL references. For instance, if we observed a value containing 'injuries over 70', we would replace this by 'injuries over', since counts like '70' are not likely to generalize between events. Finally, we add the resulting cleaned value as a label l for the event aspect. Each infobox will typically provide multiple event aspects (e.g. 'Location', 'Rail line', and 'Deaths') for an event type e_t. By processing multiple pages, an event aspect may be assigned multiple semantically related labels l. For instance, we might represent the aspect 'Injuries' using labels such as 'injuries over', 'wounded' or 'injured'. The top half of Table 1 illustrates a sample of the event aspects and labels extracted for the 'earthquake' event type using the infoboxes contained within the top 10 ranked Wikipedia pages for the query 'earthquake'. We use these aspects in our later experiments. We apply Snowball English stemmer to the labels to reduce the impact of vocabulary mismatch against the sentences being scored (terms in the sentences are similarly stemmed).

4.1.2 Generating Aspects via Crowdsourcing. The second approach that we propose is to generate the event aspects through the medium of crowdsourcing. This approach is designed to be a semi-manual alternative, which we would expect to be more effective, but requires some manual effort. Under this approach, for an event type e_t, we show crowdsourcing workers example sentences extracted from newswire articles from past events of that type and have them suggest event aspects, forming A. In particular, for each of the 10 named event types in our evaluation dataset (see Section 5 for more detail), we first extracted sentences from news articles for events of the same type that pre-date the test events.

For our later experiments, we use the TRC2 newswire corpus[10] as our source of related sentences, which contains headlines and content from over 1.8 million news articles from the Reuters

[10]http://trec.nist.gov/data/reuters/reuters.html

news agency that pre-date the events that we use later for evaluation (to avoid contaminating our experimental setting with future evidence that a summarization system would not have access to). We divided these 1.8 million news articles into sentences and indexed those sentences using the Terrier open source IR platform [29]. Next, we ranked those sentences for each of the 10 event types, using the event aspect e_t for each type as the query. To do so, we use a language model with a Dirichlet prior [12]. However, we introduce a custom document length normalization with the aim of improving ranking performance over sentences, rather than full articles. In particular, we use a Gaussian (bell) curve to promote sentences that are around the same length as a normal English sentence (of 25 words). More precisely, we set the mean (expectation) to 25 and the standard deviation to 20.

Using this model, we retrieved the top 100 sentences for each type, forming a pool of 1000 sentences to show to our crowdsourced workers. To avoid showing crowdsourced workers irrelevant content, we performed a fast manual pass over these sentences, removing those sentences from the pool that were clearly not relevant to an event type of interest.[11] This step took one assessor 3 hours to complete and removed 467 headlines, leaving 533 headlines in the pool. We then had three crowdsourced workers suggest up to four event aspect labels per sentence. The assessment interface and worker instructions are shown in Figure 3. To perform the crowdsourcing, we use CrowdFlower,[12] – an on-demand labour website – that builds on multiple existing crowdsourcing marketplaces. The unit of assessment is a single page, which contains 5 sentences to suggest labels for. We had 3 individual crowd workers (assessors) suggest labels for each sentence. Work submitted by crowdsourcing workers was subject to a 'speed trap' of 10 seconds per page, the aim being to detect automatic bots and/or users that are randomly entering labels. To avoid over-reliance on individual assessors, the maximum number of sentences a user could assess was set to 200. Additionally, since our events are largely US-centric, we restricted the geographical regions that could participate in the labeling task to only those who use English as their primary language. We paid US$ 0.10 for each set of 5 sentences labeled. The total number of sentences for which we had workers suggest labels was 533. Hence, the total number of assessed units was 1,599, i.e. 533 sentences * 3 unique assessors.

41 unique workers attempted the crowdsourced labeling task. About 15% of the crowd workers completed the maximum number of judgments (200 sentences), followed by a 'tail' of workers who completed fewer assessments, which is a typical behavior in crowdsourced tasks [3]. The average time it took the workers to enter labels for a sentence was 18 seconds, or 1 minute 33 seconds per page (comprised of 5 sentences to be assessed). This labeling task involves free-text entry. As there is no collaboration between workers, a variety of labels will be produced that will often be synonyms of one another. As such, standard quality metrics such as label agreement are less useful here, since only exact label matches are considered as agreement. Hence, we would expect lower worker agreement than for categorical labeling tasks. Indeed, for the 1^{st} label, the proportion of times that the three users suggested the same label for a sentence was 34%. The lower half of Table 1 illustrates a sample of the event aspects and labels generated for the 'earthquake' event type using the crowd.

[11] This step could also be performed via crowdsourcing, however the time needed to prepare a crowdsourced job for this task would likely take longer than it took the assessors to perform the task.
[12] https://www.crowdflower.com/

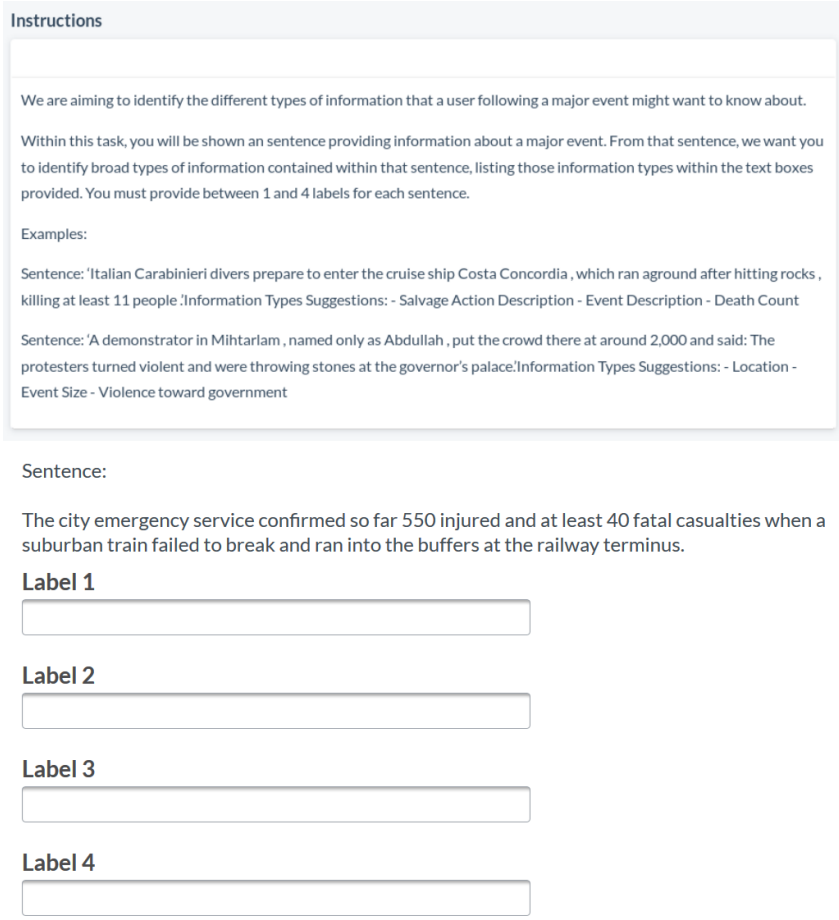


Fig. 3. Crowdsourcing instructions and interface used for event aspect labeling.

4.2 Explicit Diversification of Event Aspects

The second component of our framework is Aspect Diversification. The aim of this component is to score each update based on a set of event aspects provided by the EAG component, i.e. estimate $f(q, u, \mathrm{S})$. To this end, we adapt the state-of-the-art xQuAD explicit diversification framework [35] for temporal summarization. Within xQuAD, a candidate update $u \in \mathrm{D}$ would be scored as:

$$f(q, u, \mathrm{S}) = (1-\lambda)p(u|q) + \lambda p(u, \mathrm{S}|q), \tag{1}$$

where $p(u|q)$ denotes the probability that u is relevant given the query q and $p(u, \mathrm{S}|q)$ denotes the probability that u is diverse compared to the already selected updates in S. The mixing parameter λ balances the trade-off between relevance and diversity in the final score. The probability of diversity $p(u, \mathrm{S}|q)$ can be further decomposed by explicitly modelling the various aspects A underlying the query q, according to:

$$p(u, \bar{\mathrm{S}}|q) = \sum_{a \in \mathrm{A}} p(a|q)p(u|q, a) \prod_{v \in \mathrm{S}} (1 - p(v|q, a)), \tag{2}$$

Table 1. Example event aspects and representations for an earthquake event obtained from Wikipedia infoboxes and via crowdsourcing.

Wikipedia Infobox Event Aspects			
Event Aspect ($a \in A$)	Labels $l \in L$		
Damage	damage	million	usd
Type	fault	geology	
Magnitude	magnitude	mw	richter
PGA	peak	ground	acceleration
Casualties	killed	dead	died
Depth	depth	km	

Crowdsourced Event Aspects			
Event Aspect ($a \in A$)	Labels $l \in L$		
Deaths	death	killed	toll
Buildings	damage	destroyed	infrastructure
Strength	magnitude		
Response	government	aid	response
Location	epicentre	longitude	latitude

where $p(a|q)$ denotes the *importance* of the aspect a given q, $p(u|q,a)$ denotes the *coverage* of the update u with respect to this particular aspect, and the rightmost product denotes the *novelty* of u, in terms of how poorly this aspect is already covered by the updates v previously selected in S.

A common assumption made by explicit diversification approaches including xQuAD is that the aspects A underlying a query q are independent with respect to one another. In contrast, as introduced in Section 4.1, each of the aspects identified for the target type of q is ultimately represented by multiple labels with correlated semantics (e.g., 'epicentre', 'latitude', 'longitude'). Notably, hierarchical aspect modeling has recently been shown to provide a more accurate representation of the interdependencies among semantically correlated aspects, leading to a significant improvement in the resulting diversification effectiveness [20]. While a full hierarchical modeling of event aspects is beyond the scope of this article, we devise an extension of xQuAD in Equation (2) to model dependencies among the various labels L representing each aspect $a \in A$ underlying the query q, according to:

$$p(u, \bar{S}|q) = \sum_{a \in A} p(a|q) \sum_{l \in L} p(l|a) p(u|q,a,l) \prod_{v \in S} (1 - p(v|q,a,l)), \quad (3)$$

where the probability $p(l|a)$ denotes the importance of label l given the aspect a, whereas $p(u|q,a,l)$ and the rightmost product denote refined coverage and novelty probabilities for the update u with respect to l, respectively.

In order to estimate the various probabilities in Equation (3), we make a few assumptions. Firstly, we assume a uniform distribution $p(a|q) = 1/|A|$, $\forall a \in A$, which has been shown to be effective for web search result diversification [35]. Next, in order to estimate $p(l|a)$, we note the complementary nature of the multiple labels $l \in L$ identified for a given aspect. Accordingly, we assume that an update u should need to only match one of the labels in L that represent the aspect a for

the update to be considered to cover a. Formally, we define:

$$p(l|a) = \begin{cases} 1, & \text{if } l = \arg\max_{k \in \mathrm{L}} p(u|q,a,k), \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

so that only the label l that best supports (has the highest likelihood of matching) the current update u contributes to estimating the coverage and novelty of this update, with all other labels $k \in \mathrm{L} \setminus \{l\}$ having no impact. This helps us break ties where an update contains two labels associated to a single aspect. For the previously illustrated earthquake event, only the best supporting label associated with the aspect a would be considered in Equation (3) when assessing the coverage and novelty of update u with respect to this aspect. For instance, if $l \in$ {'deaths', 'killed', 'toll'}, and probabilities of the update u given each label were deaths=0.8, killed=0.0 and toll=0.6, then $p(u|q,a,deaths)$ would be used.

Lastly, in order to estimate $p(u|q,a,l)$, which is used in Equation (3) (to convey the coverage of an update u and the novelty of this update with respect to every already observed update $v \in \mathrm{S}$) as well as in Equation (4), we assume $p(u|q,a,l) \approx p(u|l)$ for simplicity, given that all labels are already query- and aspect-biased by construction. For instance, continuing the example above, if $q =$ '*Hurricane Isaac*', $a =$ '*fatalities*' and $l =$ '*deaths*', we know a priori that 'deaths' is a synonym of 'fatalities' and that 'fatalities' is an aspect of the event being represented by the query 'Hurricane Isaac'. With this simplification, both $p(u|q)$ in Equation (1) and $p(u|l)$ can be estimated by modelling either the query or the label generation process, respectively.

For ease of reference, we denote our explicit diversification framework for temporal summarization as xQuAS—Explicit Query Aspect Summarization. In the next section, we discuss our experimental setup, while our results are presented in Section 7.

Table 2. Event '2012 Costa Concordia Disaster'.

Event	2012 Costa Concordia Disaster
Representation (Query) q	'costa concordia'
Event Type *type*	Accident
Time Range	Start: 13 Jan 2012 21:45 End: 01 Feb 2012 00:00

5 EXPERIMENTAL SETUP

Datasets: To evaluate xQuAS, we use the TREC 2013-2015 Temporal Summarization (TREC-TS) task datasets. The track used the TREC Knowledge-Based Acceleration (KBA) stream corpus to represent the documents being published over time, for each year respectively. This corpus contains over 1 billion timestamped Web documents (e.g. news articles, blogs and forum posts) from the period of October 2011 to April 2013.[13]

Events: Each TS task dataset has a series of topics that correspond to large events, along with a user query q representing each. The TS 2013 task used 9 topics (events), the 2014 task used 15 topics (events), while the TS 2015 task used 21 topics (events), for a total of 45 topics. The topic definition for each event also provides an event type, e.g. 'storm' or 'earthquake', which we use to match an event to our event aspects A (see Section 4.1). There are 10 event types in total, namely: Accident; Shooting; Storm; Earthquake; Bombing; Cold Wave; Flood; Riot; Protest; and Impact Event. Each event has a pre-determined time range. For an event, summarization systems

[13] http://trec-kba.org/

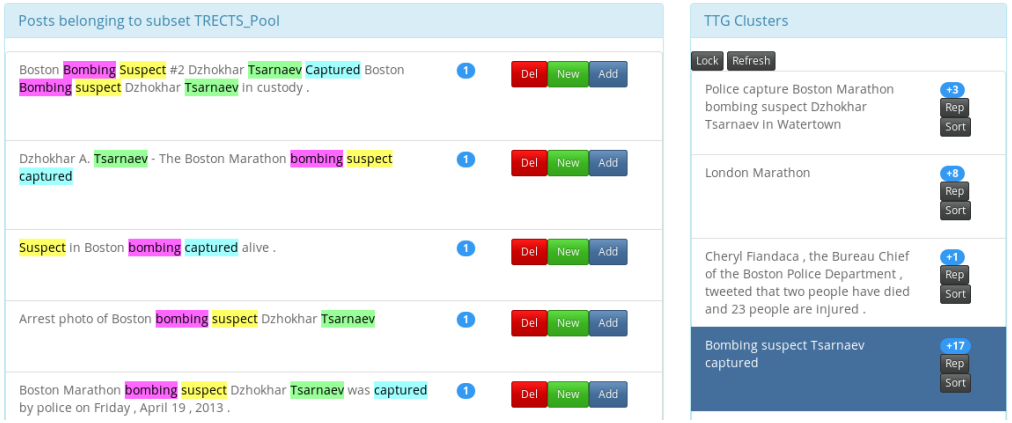


Fig. 4. TREC-TTG clustering interface.

simulate the processing of documents in time-order from the KBA corpus during the time-frame of that event, emitting updates into the summary as they are identified. Table 2 illustrates an example event.

Stream Processing: Notably, in order to focus the corpus on documents that are more likely to contain relevant content for each event, the track organizers provided the participants with a pre-filtered version of the KBA corpus for the 2013-2015 events that removes documents from outside the time ranges of the those events and applies basic keyword filtering, which we use here. Furthermore, in addition to the track filtering, we also apply additional pre-processing to the document stream. First, documents that do not contain one or more of the event query (q) terms are also filtered out (this avoids processing documents that will never have any sentences selected from them). Second, basic sentence level filtering is applied, which removes very short sentences (those containing less that 10 terms.[14] All of the approaches tested in our experiments follow the baseline summarization framework described in Section 3. As such, sentences extracted from the stream are buffered into pre-defined time intervals. At the end of each interval, one of the selection criteria is then applied to each sentence within that interval. We use a time interval of 1 hour, as this has previously been shown to be effective [28].

Baselines: To evaluate whether summary diversification based on explicit event aspects is more effective than relying on novelty-based methods, we compare our approach to baselines that use implicit diversification, generated using the basic temporal summarization framework defined in Section 3. Recall that under the basic temporal summarization framework, approaches are comprised of three parts: the scoring function; the selection criterion; and the redundancy removal technique. For our later experiments, we test two different scoring functions:

(1) The first scoring function we use is relevance of the update to the event query. More precisely, we use a language model with a Dirichlet prior to calculate $p(u|q)$ [50].

[14] We choose a relatively strict filtering here, as very short updates often lack sufficient context to be interpretable by a user. Additionally, the sentence extraction algorithm applied to the KBA corpus means that there are many 5-8 word sentences that are article boilerplate. Setting a 10 term threshold avoids processing these types of sentences.) and those sentences that contain non-English characters.

(2) The second scoring function we implement is the salience-based scoring function proposed by Kedzie et al. [22]. We then combine it with either the *topk* or *threshold* selection criterion, and apply implicit diversification via greedy cosine redundancy removal, as with the first baseline type.[15] The salience-based scoring function is a trained regression scorer that combines features of the sentences (e.g. sentence length), query features (e.g. similarity of the sentence to the event query), language model features (e.g. similarity of the sentence to a language model built for each event type), geographic features (distance to the predicated location of the event) and Temporal Features (does the sentence contain words that are appearing more often than normal). We follow the methodology and implement the features described by Kedzie et al. [22]. However, to maintain a consistent learning configuration, we use the 5-fold setting discussed later, rather than the leave-one-out setting used in the original article. Furthermore, as the original article does not describe the methodology for obtaining geo-locations from each document, we choose to use the GATE platform's [13] ANNIE pipeline to identify location mentions within each document and the Geonames[16] service to lookup coordinates for each location.

For both scoring functions, we combine them with *topk* and *threshold* selection criteria and classical redundancy removal based on cosine similarity. In particular, for each scoring function, we generate two *topk* selection baselines, where k is 1 or 3, and 9 baselines that use *threshold* selection, where the threshold $\tau = [0.1, 0.2, 0.3, ..., 0.9]$. In this way, we generate 22 baselines, 11 based on the relevance scoring function and 11 based on the salience-based scoring function. All of these baselines are included in the pool we use later for evaluation (see Section 6).

Evaluation Metrics: We evaluate using the TREC-TS track official metrics: Expected Gain; Comprehensiveness; and Latency Discount [16]. Expected Gain measures for each update (sentence) issued to the user, whether that update belongs to an information cluster that has not been covered by a previously returned update. This metric measures summary precision, where precision is defined as the proportion of updates that contain new information (nuggets). In contrast, comprehensiveness measures the coverage of an event based on all of the information nuggets for that event, i.e. it measures the summary recall. The Latency Discount measures the timeliness of the information returned in the summary with respect to when that information was first observed within the document stream, discounting the score for updates containing outdated information. Note that due to the way the Latency Discount is calculated, a higher Latency Discount is better (higher values mean less reporting latency). For clarity, in our later experiments, we refer to this metric as Timeliness (again, higher is better). We also report the TREC track's official combined metric, which is the harmonic mean of Expected Gain and Comprehensiveness, each mixed with the Latency Discount. We denote this metric as *Combined*. Metric formulations can be found in Appendix B.

Parameters and Training: For our proposed scoring function that is based on the xQuAD diversification framework, we set the relevance/novelty tradeoff parameter λ to 0.5, following Santos et al. [35]. In addition, we consider a second instantiation of our approach with $\lambda = 1.0$, which is equivalent to using a hierarchical version of IASelect [1] as the basis for diversification [37]. For our experiments, we use the scores produced by our approaches in conjunction with the *threshold* selection criteria. When reporting temporal summarization performances for approaches that use *threshold*-based selection, we train the threshold τ using a 5-fold cross validation where 9

[15] It is worth noting that Kedzie et al. [22] used a different selection criterion based on affinity propagation in their work. However, to avoid introducing an additional confounding variable and to maintain a consistent setup, we only implement the salience-based scoring part of their model.

[16] http://www.geonames.org

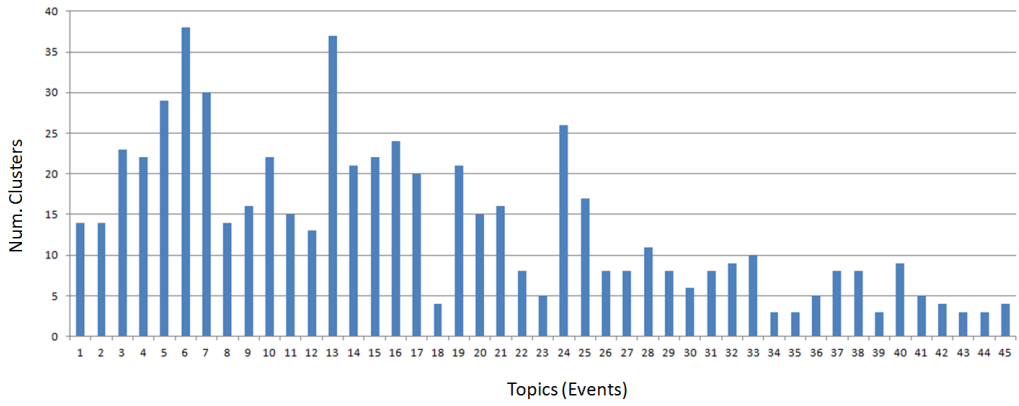


Fig. 5. Number of clusters per topic.

topics/events are assigned to each fold. We define our folds based on event boundaries to avoid training on data from the same event/time period as the test data. We report the average performance over the test topics for each fold, when training for the *Combined* metric. In the case of the second baseline approach that uses a learned regression model for scoring, following Kedzie et al. [22] we infer training labels for each event by calculating ROUGE-2 F1 similarity scores between each sentence to be scored for that event and the gold-standard 'information nuggets' provided as part of the TREC-TS dataset for the event. We then divide the labeled training instances into the same 5-folds as used for threshold training. For each fold, sentence score predictions are made based on a model trained on the other 4 folds.

6 EVALUATION AND LABELING

Along with the datasets for each year of the track, TREC-TS also provides a pool of assessed updates to enable the assessment of additional summarization systems. However, as noted in other studies, these assessments are too incomplete to be reusable when developing novel systems that are markedly different from those that originally participated in the track [21, 30]. To illustrate this is a problem for the approaches we proposed in this work, we include a comparison of performances and assessment completeness for the approaches proposed here in comparison to the TREC top systems for the first year of the track (2013) in Appendix A. Since we cannot use the original assessments produced by TREC out-of-the-box, we generate different ground truth label sets, denoted 'TTG-All', 'TTG-201X' and 'TRECTS-201X', as described below.

6.1 'TTG-All' Label Set

To evaluate the baseline and proposed summarization systems, we adopt the evaluation methodology used by the TREC Microblog track Twitter Timeline Generation (TREC-TTG) task. The TREC-TTG assessment methodology is a variant of that used by the TREC-TS track,[17] where the top scoring updates returned from each summary per-event are first pooled, and then manually clustered by human assessors based on the information that they contain. For instance, the updates 'Breaking News: 21 people have been confirmed dead in a building collapse in Dhaka' and

[17] Note that we do not reproduce TREC-TS track assessment methodology here, as initial testing indicated that the cost of producing TREC-TS track assessments (update to nugget labels) is between 3-4 times higher than producing TREC-TTG assessments (update to cluster labels).

'Bangladesh, Dhaka: Eight-story commercial building collapsed, reports indicate 21 dead and many missing' might be clustered together based on the fact that they both mention the building collapse and give the same number of fatalities. Following the TREC-TTG methodology, we pooled the top 60 scoring updates from each system summary per event. We then employed a set of 6 assessors (students and researchers) to cluster the updates within the pool for each event. Each assessor was assigned a subset of the events, and could create any number of information clusters per event. For each update, an assessor could add that update to an existing cluster, define a new cluster for it, or delete the update (if it was non-relevant or otherwise not suitable for inclusion in a summary). The interface used to cluster the updates is shown in Figure 4.

In total, we pooled the top 60 updates from 22 baseline runs (11 based on relevance scoring and 11 based on the learned salience model [22]) that use implicit diversification and 36 runs that use either the infobox or crowdsourced explicit diversification labels. The number of unique updates in the pool is 6,635. It took approximately 32 hours spread out over the 6 assessors to cluster these updates for the 46 topics (events). Of the 6,635 updates pooled, 4,197 (63%) were judged relevant (were added to one or more clusters). Figure 5 shows the number of clusters defined per topic (event). As can be seen from Figure 5, the largest number of clusters for an event was 38 (topic 6, Hurricane Sandy), while the smallest number of clusters for an event was 3 (e.g. topic 44, 2012 Indian Ocean Earthquakes). We refer to this label set as the *'New-TTG'* label set. This label set covers all 45 topics used during TREC-TS 2013-2015 and is the primary means by which we will evaluate the proposed approaches.

6.2 'TRECTS-201X' and 'TTG-201X' Label Sets

On the one hand, it may be useful to compare against the state-of-the-art systems that participated during the three years of the TREC-TS track. On the other hand, the 'TTG-All' label set is not suitable for evaluating the participating systems to the TREC-TS track for the same reason that we could not use the original TREC-TS ground truth assessments to evaluate our new proposed approaches, i.e. there is insufficient completeness in the ground truth labels (see Appendix A for an illustration of this issue). Moreover, the systems that participated in TREC-TS varied from year-to-year, hence, we can only compare by-year, instead of across years (as we do with the 'TTG-All' label set.

To enable a comparison, we first split the 'TTG-All' label set into three separate event sets, where each contains the events used for evaluation during the three years of the TREC-TS track. More precisely, events 1-10[18] are assigned to a 'TTG-2013' label set, events 11-25 to a 'TTG-2014' label set and events 26-46 to a 'TTG-2015' label set. We use these label sets later to provide an approximate comparison of the performance of our proposed approaches to the TREC best participating systems for each year. We do this by reporting the performance of the top three TREC-TS systems each year when evaluated on the official TREC-TS assessments (denoted 'TRECTS-2013', 'TRECTS-2014' and 'TRECTS-2015' respectively), in comparison to our proposed approaches when evaluated on the associated TTG-equivalent. For example, the TREC-TS top 3 systems for 2013 are evaluated on the 'TRECTS-2013' label set and compared to our proposed approaches evaluated on the 'TTG-2013' dataset. However, it should be stressed that this comparison should only be considered as approximate. In particular, the labeling methodologies used to create the nuggets and matches, the interfaces and support tools used to do the matching, as well as the assessor profiles differ between the TREC-TS original assessments ('TREC-TS-201X' label sets) and the label sets derived from 'TTG-All' ('TTG-201X' label sets). For those interested in examining the differences between these methodologies in more detail, we recommend reading the study by Baruah et al. [4].

[18]Event 7 is ignored as per the TREC-TS 2013 evaluation.

7 RESULTS

In this section we investigate four research questions:

- **RQ1**: How effective is classical novelty-based diversification of temporal summaries? (Section 7.1).
- **RQ2**: Does explicit diversification based on summarization aspects enhance the quality of the summaries? (Section 7.2)
- **RQ3**: Which of the two event aspect generation approaches described in Section 4.1 is more effective? (Section 7.3)
- **RQ4**: Where do explicit summary diversification approaches fail? (Section 7.4)

7.1 Implicit Diversification Performance

Initially, it is important to determine what an effective baseline performance is for our evaluation scenario, to form a basis for later comparisons. Hence, we first evaluate the performance of different baseline summarization approaches that make use of classical implicit diversification approaches. Recall from our baseline discussion that implemented two update scoring functions, one based on relevance to the event query Q (which we denote as *Relevance*), and a learned model that estimates update salience (that we denote as *L-Salience*), based on prior work by Kedzie et al. [22]. Following McCreadie et al. [30], we first combine these scoring functions with the *topk* selection criteria, where k is 1 or 3, denoted *Top1* and *Top3*, respectively. Second, we similarly combine the two scoring functions with $threshold$-based selection, where the threshold τ has been trained using a 5-fold cross validation (the reported performance in this case is the average performance across the test fold of each round), denoted *Threshold*. For all of these baselines, we apply implicit diversification by filtering out updates that are textually similar (cosine similarity, threshold τ=0.7) to those already selected (denoted *Novelty*). Table 3 reports the performance of these 3 baseline approaches under the TREC-TS/TREC-TTG evaluation metrics, i.e. *Expected Gain*, *Comprehensiveness*, *Timeliness* and the *Combined* metric using the 'TTG-All' label set. In all cases, higher scores are better. For illustration, we also report the performances of the best systems that participated in the TREC-TS track during 2013, 2014 and 2015 in comparison to our baselines in Table 4. In this case, performances are calculated per-year, rather than across all years, using the 'TRECTS-201X' and 'TTG-201X' label sets, respectively.

From Table 3, we observe the following. First, examining the two *topk* baselines, we see that when using *Relevance* scoring, *topk* selection naturally leads to summaries that perform well in terms of comprehensiveness, i.e. they contain most of the relevant information about the event. However, this comes at the cost of expected gain (precision), i.e. many of the updates returned are redundant or non-relevant. This can be explained by the fact that many of the events are long running, and there is not always new content to return during each hour time interval, as reported previously by McCreadie et al. [30]. Hence, during some time intervals the top documents are redundant or non-relevant. As a result, the *Relevance topk* baselines perform poorly overall. Meanwhile, considering the *L-Salience topk* baselines, we see that expected gain (precision) is significantly higher, but comprehensiveness (recall) is significantly lower. These scores can be explained by the fact that *L-Salience topk* produces very short summaries for each event, where the content returned is often relevant, but incomplete. As a result, the overall performance of *L-Salience topk* baselines is higher than the *Relevance topk* baselines.

Next, we compare the *Relevance* baseline that uses the score threshold selection strategy (*Threshold*) to their *topk* counterparts. From the results, we observe that $threshold$ selection is more effective when using the *Relevance* scorer than *topk* with a Combined score of 0.2104 (Relevance+Threshold) vs. 0.0893 and 0.0640 (*Relevance+Top1* and *Relevance+Top3*, respectively). Indeed, examining this

Table 3. Baseline Temporal Summarization performances over the 46 TREC-TS topics (events). The best performing baseline under each measure is highlighted in bold. Statistically significant increases/decreases (paired t-test $p<0.05$) in performance over the *Top 1* approach are denoted ▲ and ▼, respectively.

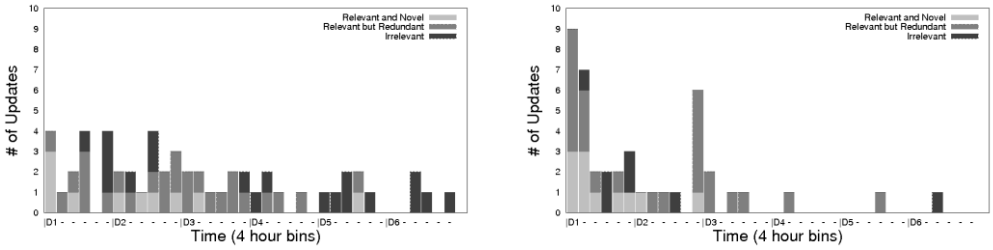
TREC-TS 2013-2015, TTG-All (46 Events)							
				TREC TS Metrics			
Scoring	Aspects	Selection	Redundancy	Expected Gain	Comprehensiveness	Timeliness	Combined
Relevance	None	Top1	Novelty	0.0765	0.7723	0.6536	0.0893
Relevance	None	Top3	Novelty	0.0441▼	**0.8275**▲	*0.7623* ▲	0.0640▼
L-Salience	None	Top1	Novelty	0.1337▲	0.2813▼	0.3263▼	0.1812▲
L-Salience	None	Top3	Novelty	0.0641	0.3275▼	0.4969▼	0.1072▲
Relevance	None	Threshold	Novelty	0.3083▲	0.6026▼	0.5823▼	0.2104▲
L-Salience	None	Threshold	Novelty	0.0947▼	0.2378▼	0.3765▲	0.1355▼

Table 4. Temporal Summarization performances under the Combined metric for the baselines and TREC best systems for each of the three years of the TREC-TS track. The TREC-TS top three systems are evaluated on the 'TRECTS-201X' label sets, while the baseline approaches are evaluated against the 'TTG-201X' label sets. Comparison is approximate only, see Section 6.2 for caveats to consider when viewing these performance numbers. The best performing system for each year is highlighted in bold.

	Combined		
Run	TRECTS-2013 (9 Events)	TRECTS-2014 (15 Events)	TRECTS-2015 (21 Events)
TREC-TS Rank 1	0.2311 (ICTNET/run1)	0.3221 (BJUT/Q1)	0.2818 (UWCTS/Run1)
TREC-TS Rank 2	0.1694 (PRIS/cluster2)	0.2301 (cunlp/2APSal)	0.2696 (CWI/IGnPrecision)
TREC-TS Rank 3	0.1213 (uogTr/NMTm1MM3)	0.2156 (uogTr/2A)	0.2140 (cunlp/3LtoSfltr5)
TREC Median	0.0708	0.0907	0.1493

Scoring	Aspects	Selection	Redundancy	TTG-2013 (9 Events)	TTG-2014 (15 Events)	TTG-2015 (21 Events)
Relevance	None	Top1	Novelty	0.1526	0.174	0.1054
Relevance	None	Top3	Novelty	0.0839	0.1314	0.0480
L-Salience	None	Top1	Novelty	0.1206	0.0917	0.1512
L-Salience	None	Top3	Novelty	0.1577	0.0795	0.0867
Relevance	None	Threshold	Novelty	0.3494	0.4168	0.3792
L-Salience	None	Threshold	Novelty	0.1883	0.0343	0.0851

result more closely, we see that the superior performance of the threshold-based approach is due to much better Expected Gain (precision), e.g. *Relevance+Threshold*:0.3083 vs. *Relevance+Top1*:0.0765. In practical terms, this means that the threshold-based selection returns fewer irrelevant or redundant updates. However, it should be noted that this does come at the cost of some Comprehensiveness (recall), e.g. *Relevance+Threshold*:0.6026 vs. *Relevance+Top1*:0.6536. To illustrate why this is the case, we compare the updates returned by the *Relevance+Top1* and *Relevance+Threshold* approaches for a single topic. To do this, we introduce the concepts of relevant and novel updates, relevant but redundant updates and non-relevant updates. A relevant and novel update returned for an event is one that was (manually) matched (see Section 6) to one or more of the gold-standard information nuggets extracted for that event, and where one or more information nuggets matched have not been covered by previously selected updates (i.e. the update contains some new information). A relevant but redundant update is one that matches one or more information nuggets, but all of the matched nuggets have already been covered by other updates returned beforehand. An irrelevant update is one that did not match any of those information nuggets. Figure 6 shows the number of relevant and novel (light grey), relevant but redundant (mid-tone grey) and irrelevant (dark grey) updates returned by the *Top1* (Figure 6 (a)) and *Threshold* (Figure 6 (b)) approaches for event 10: the 2012 Tel Aviv bus bombing. The x-axis in these figures represents the duration of that



(a) Top1+Novelty (b) Threshold+Novelty

Fig. 6. Visualization of the number of relevant (light grey) and non-relevant (dark grey) updates returned by the Top1+Novelty and Threshold+Novelty approaches over time for event 10 (2012 Tel Aviv bus bombing). Each column represents a 4-hour period.

event, where each column describes a 4-hour period. i.e. the first column represents updates returned during the first 4 hours of the event, the second column represents updates returned during hours 5-8, and so on. It should be noted that for the *Relevance+Top1* approach, the maximum number of updates that it can return in any 4-hour period is 4 (one per hour). It may return less than that in cases where the textual novelty-based redundancy removal (which is applied to both types of approach) filters out one or more of those updates. Comparing Figure 6 (a) and Figure 6 (b), we observe that the *Relevance+Top1* approach returns more irrelevant updates during the second-half of the event life-time where there is little or no new relevant content available. This contrasts to the *Threshold+Novelty* approach which avoids returning any updates during those times.

Third, comparing the *Relevance+Threshold* baseline to the *L-Salience+Threshold* baseline, we see that using *Relevance* scoring is more effective (*Relevance+Threshold*:0.2104 vs. *L-Salience+Threshold*:0.1355. As we can see from the individual metrics, this is due both to lower expected gain and lower comprehensiveness. Lower comprehensiveness is to some extent expected, as we noted during the *topk* discussion that *L-Salience*-based scoring leads to short summaries that miss information. However, the lower expected gain also indicates that *L-Salience* is not effectively placing relevant updates in the top ranks for each hour. While it may seem counterintuitive for the unsupervised model (*Relevance*) to outperform a learned combination (*L-Salience*), it has been previously observed that effective learned models are difficult to construct for this task [30]. Indeed, based on past experience, the authors surmise that this poor performance is likely due to a combination of the ROUGE-based training labels not accurately reflecting update relevance/quality and the learner being misled by term usage patterns that can vary greatly between events of different types.

Overall, to answer RQ1, we conclude that *Relevance* scoring combined with threshold-based selection along with textual-novelty for redundancy removal (*Relevance+Threshold*) is significantly more effective than *topk* alternatives that also use textual-novelty for redundancy removal, with a performance under the *Combined* metric of 0.2104. The reason for the better performance of the threshold-based approach is a marked reduction in the volume of irrelevant updates returned later on in an event's lifetime. As the *Relevance* and *Threshold* combination is more effective overall, we use it as the baseline in our analysis of explicit diversification approaches in the next section.

7.2 Explicit Diversification Performance

Having determined what a baseline performance is in this scenario when using implicit diversification, we next examine the performance of approaches that use the explicit event aspect labels

Table 5. Temporal Summarization performances when using explicit event aspects over the 46 TREC-TS topics (events). The best performing approach under each measure is highlighted in bold. Statistically significant increases/decreases (paired t-test $p<0.05$) in performance over the Threshold+Novelty baseline are denoted ▲ and ▼, respectively.

TREC-TS 2013-2015, TTG-All (46 Events)							
				TREC TS Metrics			
Scoring	Aspects	Selection	Redundancy	Expected Gain	Comprehensiveness	Timeliness	Combined
L-Salience	None	Top1	Novelty	0.1337	0.2813	0.3263	0.1812
Relevance	None	Threshold	Novelty	0.3083	*0.6026*	0.5823	0.2104
IASelect [1]	Wikipedia	Threshold	None	**0.7955**▲	0.2148▼	**0.7735**▲	0.2533▲
IASelect [1]	Crowd	Threshold	None	0.6519▲	0.2584▼	0.7279▲	0.2660▲
xQuAS [35]	Wikipedia	Threshold	None	0.3684▲	0.4450▼	0.7190▲	0.2598▲
xQuAS [35]	Crowd	Threshold	None	0.5369▲	0.3252▼	0.7807▲	**0.3124**▲

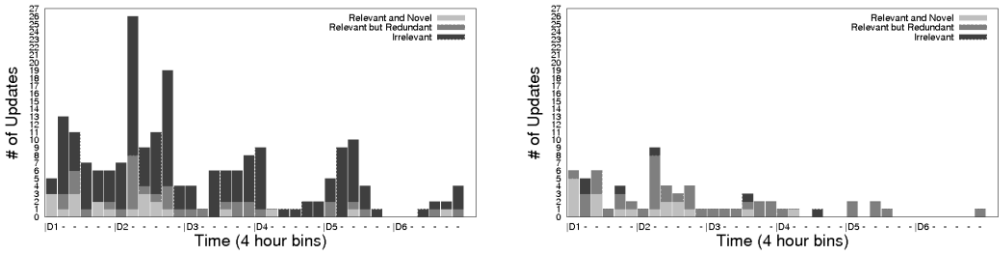
Table 6. Temporal Summarization performances under the Combined metric for the proposed approaches that use explicit event aspects and TREC best systems for each of the three years of the TREC-TS track. The TREC-TS top three systems are evaluated on the 'TRECTS-201X' label sets, while the 'proposed approaches are evaluated against the 'TTG-201X' label sets. Comparison is approximate only, see Section 6.2 for caveats to consider when viewing these performance numbers. The best performing system for each year is highlighted in bold.

	Combined		
Run	TRECTS-2013 (9 Events)	TRECTS-2014 (15 Events)	TRECTS-2015 (21 Events)
TREC-TS Rank 1	0.2311 (ICTNET/run1)	0.3221 (BJUT/Q1)	0.2818 (UWCTS/Run1)
TREC-TS Rank 2	0.1694 (PRIS/cluster2)	0.2301 (cunlp/2APSal)	0.2696 (CWI/IGnPrecision)
TREC-TS Rank 3	0.1213 (uogTr/NMTm1MM3)	0.2156 (uogTr/2A)	0.2140 (cunlp/3LtoSfltr5)
TREC Median	0.0708	0.0907	0.1493

Scoring	Aspects	Selection	Redundancy	TTG-2013 (9 Events)	TTG-2014 (15 Events)	TTG-2015 (21 Events)
L-Salience	None	Top1	Novelty	0.1206	0.0917	0.1512
Relevance	None	Threshold	Novelty	0.3494	0.4168	0.3792
IASelect	Wikipedia	Threshold	None	0.4541	0.4108	**0.4076**
IASelect	Crowd	Threshold	None	0.3665	0.3213	0.3751
xQuAS	Wikipedia	Threshold	None	0.5313	**0.4233**	0.3651
xQuAS	Crowd	Threshold	None	**0.5721**	0.3651	0.3620

that were either extracted from Wikipedia infoboxes or were generated via crowdsourcing (in Section 4.1). Table 5 reports the temporal summarization performance in terms of the TREC metrics of approaches that use either the proposed xQuAS or IASelect scoring formulations in comparison to the best baseline system identified in the previous section (*Threshold+Novelty*). As with the threshold-based baseline, we train the selection threshold of the IASelect and xQuAS runs using a 5-fold cross validation. The best system under each metric is highlighted in bold. Statistically significant increases/decreases (paired t-test $p<0.05$) over the best baseline approach (identified previously in Section 7.1) are denoted ▲ and ▼, respectively. Additionally, for illustration, we also report the performances of the best systems that participated in the TREC-TS track during 2013, 2014 and 2015 in comparison to our baselines in Table 4. In this case, performances are calculated per-year, rather than across all years using the 'TRECTS-201X' and 'TTG-201X' label sets, respectively. For easy reference in this section, we denote the approaches that explicitly diversify in the following format: Scoring+Aspects. For example, the approach that uses xQuAS for scoring with aspects extracted from Wikipedia infoboxes is denoted *xQuAS+Wikipedia*.

From Table 5, we make the following observations. First, considering the performance of the xQuAS-based approaches under the Expected Gain metric, we see that performance is statistically significantly higher in all cases than the baseline approach that uses implicit diversification. This



(a) Threshold+Novelty (b) xQuAS+Crowd

Fig. 7. Visualization of the number of relevant and novel (light grey), relevant but redundant (mid-tone grey) and non-relevant (dark grey) updates returned by the Threshold+Novelty and xQuAS+Crowd approaches over time for event 5 (Hurricane Isaac). Each column represents a 4-hour period.

indicates that the use of pre-generated event aspects within the update scoring process results in an overall higher proportion of relevant and non-redundant updates being returned. Indeed, this is intuitive, since for the score of an update to reach the selection threshold, that update will need both a high relevance score for the query and a high novelty score with respect to one or more of the event aspects. To illustrate this behavior, Figure 7 shows the distribution of relevant and novel, relevant but redundant, and irrelevant updates returned by the baseline *Threshold+Novelty* approach (Figure 7 (a)) in comparison to the *xQuAS+Crowd* approach (Figure 7 (b)) for event 5 (Hurricane Isaac). As can be seen from Figure 7, the baseline (*Threshold+Novelty*) returns far more irrelevant updates for event 5 than the *xQuAS+Crowd* approach.

Next, from Table 5, examining the performance of the explicit diversification approaches under the Comprehensiveness metric, we observe that summary comprehensiveness is lower by a statistically significant margin in all cases against the baseline (see Section 7.1). To explain this behavior, we need to consider the event aspects that we are diversifying for. In particular, recall that we generate our event aspects prior to the start of the event from other past events of the same type (to avoid using evidence from the future). A consequence of this design choice is that the event aspects we generate are quite generic and hence do not always capture information that is unique to each event. For instance, consider the 2012 Costa Concordia Disaster (with event type 'Ship Disaster'), where a cruise liner ran aground near the island of Isola del Giglio, Italy. For this event, our generated event aspects cover general information such as the number of deaths and injuries, the location and details of the rescue operation. However, for this event, the update clustering performed by the assessors resulted in information clusters dedicated to information such as accusations against the ship captain for evacuating early, or the subsequent lawsuit against the company that runs the cruise ship. As these types of information are unique to this particular event, the event aspect generation processes described in Section 4.1 cannot easily capture them. This is the core reason that the explicit diversification approaches provide lower summary comprehensives on average over the different events than the Relevance baseline.

Next, by examining the Latency column in Table 5, we see that the xQuAS-based approaches outperform (i.e. have better Timeliness) the baseline by a statistically significant margin. This increase in timeliness in comparison to the baseline approach can be explained by the introduction of the event aspects into the update scoring function. In particular, as we previously observed in the baseline analysis in Section 7.1, not all of the information that we want to return have high relevance scores. By mixing these relevance scores with the event aspect novelty scores, we increase the likelihood that we will score highly the first case of an update containing information

Table 7. The first six updates returned by the Relevance+Threshold baseline and the xQuAS+Crowd system for Event 11: 'Costa Concordia Disaster'.

Run	Timestamp	Update Text
Relevance+Threshold	01/14/2012, 5:02pm	Carrying 3,206 passengers and 1,023 crew members, the Costa Concordia was on its usual weekly route across the Mediterranean Sea and departed Civitavecchia - the port of Rome - three hours before disaster struck.
Relevance+Threshold	01/14/2012, 9:38pm	As the Costa Concordia keeps shifting on its rocky ledge, many have raised the prospect of a possible environmental disaster if the 2,300 tons of fuel on the half-submerged cruise ship leaks into the sea.
Relevance+Threshold	01/15/2012, 2:04am	Jan. 16 (Bloomberg) – Carnival Corp.'s losses following the grounding of the Costa Concordia cruise ship off Italy may be exacerbated by the disaster coinciding with the start of the peak booking season.
Relevance+Threshold	01/15/2012, 5:17pm	The Costa Concordia death toll has risen by two - as all British passengers and crew were confirmed to have survived the disaster.Two French nationals and a Peruvian died after the Italian - owned cruiser ran aground near the island of Giglio off the Tuscan coast on Friday night.
Relevance+Threshold	01/15/2012, 7:41pm	Naufragio Costa Concordia , parlano i sommozzatori - Costa Concordia ship disaster Proseguono senza sosta le ricerche di passeggeri rimasti intrappolati nella costa concordia adagiata sugli scogli dell'Isola del Giglio.
Relevance+Threshold	01/15/2012, 7:03pm	search for 40 missing passengers in Costa Concordia cruise ship disaster / Laurie Willits from Ontario, who was watching a magic show with her husband at that moment , said : " We heard a scraping noise to the left of the ship and then my husband said "we're sliding off our seats."
xQuAS+Crowd	01/14/2012, 6:04am	In the chaotic aftermath of the Friday evening accident near the island of Giglio off the coast of Tuscany , Italian officials could still not say how many of the 4,229 passengers and crew on board the 114,500-tonne Costa Concordia were missing.
xQuAS+Crowd	01/14/2012, 5:30am	The luxury cruise ship Costa Concordia leans after it ran aground off the coast of Isola del Giglio island , Italy , gashing open the hull and forcing some 4,200 people aboard to evacuate / Picture : AP
xQuAS+Crowd	01/14/2012, 5:20am	Two people were killed after a luxury cruise ship carrying more than 4,000 ran aground off the Italian coast Friday , prompting a major rescue operation that is still underway.The Costa Concordia had left the port of Savona at 7 p.m. at the start of a seven-day cruise of the Mediterranean.
xQuAS+Crowd	01/14/2012, 5:15am	Forty people from the stricken cruise liner Costa Concordia are still missing as it was reported that the ship 's captain and first officer were being questioned by Italian prosecutors on suspicion of manslaughter.
xQuAS+Crowd	01/14/2012, 5:02pm	Carrying 3,206 passengers and 1,023 crew members , the Costa Concordia was on its usual weekly route across the Mediterranean Sea and departed Civitavecchia - the port of Rome - three hours before disaster struck.
xQuAS+Crowd	01/15/2012, 5:17pm	The Costa Concordia death toll has risen by two - as all British passengers and crew were confirmed to have survived the disaster.Two French nationals and a Peruvian died after the Italian - owned cruiser ran aground near the island of Giglio off the Tuscan coast on Friday night.

for each event aspect, thereby decreasing summarization latency for information clusters relating to those event aspects. To illustrate this, Table 7 shows the first six updates returned by the *Relevance+Threshold* baseline and the *xQuAS+Crowd* approach for event 11: Costa Concordia Disaster. From Table 7 it can be seen that the explicit *xQuAS+Crowd* approach returns more updates toward the beginning of the event (4 updates returned from the 14th of January as opposed to only 2 updates by the *Relevance+Threshold* approach), including information about the number of people killed (one of the event aspects suggested by the crowd).

Finally, if we consider the Combined measure that incorporates all three of the TREC metrics (expected gain, comprehensiveness and latency), we see that all of the xQuAS and IASelect-based approaches outperform the best baseline system (Threshold+Novelty) by a statically significant margin. This indicates that when trying to produce update summaries that balance these three factors, explicit diversification using event aspects is more effective than relying on implicit novelty-based diversification. Hence, to answer RQ2, explicit diversification of event aspects can improve temporal summary quality over using implicit diversification techniques.

7.3 Comparing Event Aspect Representations

Having shown that approaches that explicitly diversify using event aspects are more effective than classical implicit diversification approaches for temporal summarization, we next examine how summarization performance differs between our two different event aspect generation approaches, i.e. we compare the automatic Wikipedia infobox extraction method to the crowdsourced alternative that requires manual labeling. Using an automatic method for generating event aspects such as the Wikipedia infobox extraction is advantageous due to its low cost in comparison to having human annotators generate those aspects, but may be less effective.

Table 5 reports the temporal summarization performance of the approaches that explicitly diversify using the Wikipedia infobox-based event aspects and the crowdsourced event aspects, over the 2013-2015 temporal summarization events. Comparing the approaches that use the Wikipedia infobox derived aspects to those that used the Crowd aspects in Table 5, we observe the following. Under the Combined metric, we see that the Crowd derived aspects are more effective than the Wikipedia infobox-derived ones, particularly when using xQuAS-based scoring. Indeed, using xQuAS scoring with the Crowd aspects (*xQuAS+Crowd*) achieves a score under the Combined metric of 0.3124, in contrast to 0.2598 when using the Wikipedia infobox-based aspects (*xQuAS+Wikipedia*).

To examine the differences between these two approaches more closely, Figure 8 shows the performance distribution of these two approaches per topic under the Combined metric. The top half of Figure 8 shows the *xQuAS+Crowd* performances, while the bottom half shows the *xQuAS+Wikipedia* performances. From Figure 8, we make the following observations. First, from Figure 8, we see that there are four events (19, 24, 31, and 43) where both approaches provide (close to) 0 performance. These are events where no updates were returned due to relevant content being difficult to find given the initial query q, rather than due to the event aspects (the baseline approaches, including *Threshold+Novelty* also fail for these events). Second, analyzing the remaining events we see that xQuAS using the crowdsourced aspects performs more consistently well across those events than the same model using Wikipedia-based aspects. More precisely, using the crowdsourced labels, performance across events is 0.1 or better under the Combined metric with 4 exceptions (events 6, 18, 20, 23). Contrast this to the same model using the Wikipedia-based aspects, where 8 events have less than 0.1 performance under the Combined metric (events 5, 11, 14, 18, 20, 27, 35, 36). Very low performances for a particular event indicates that the generated aspects for that event's type were not sufficiently descriptive, leading to relevant content being

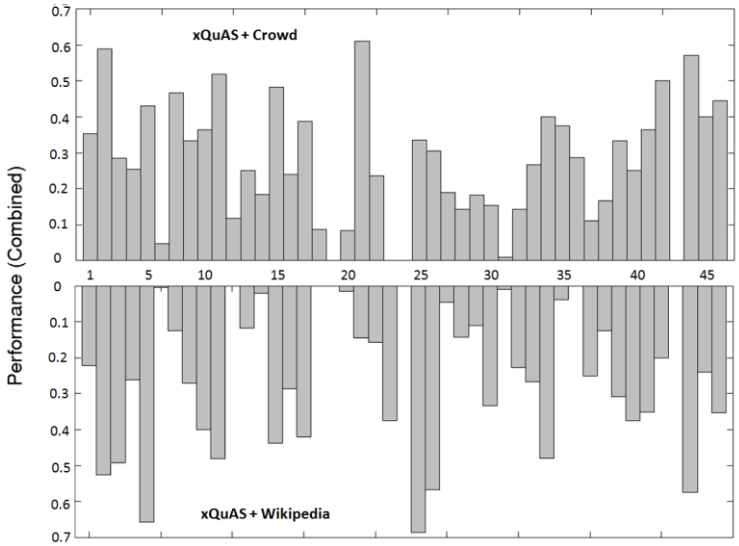


Fig. 8. Performance of the temporal summaries produced by the XQuAS + Crowd and xQuAS + Wikipedia approaches per topic under the Combined metric.

missed. Hence, our result would indicate that the crowdsourced labels produced are more descriptive than the Wikipedia-based labels (at least for the event types covered by the TREC-TS 2013-2015 datasets). Hence, to answer RQ3, we conclude that crowdsourced aspects are more effective overall than the Wikipedia-based aspects, although it should be noted that the Wikipedia-based aspects are both cheaper to obtain and do provide better performance for some events.

7.4 Failure Analysis

Finally, to provide some additional insight into where these approaches fail (and hence how we might improve them in the future) we analyze 4 poorly performing events under xQuAS when using the crowd labels. These 4 events are: Hurricane Sandy; 2011-13 Russian protests; 2012-13 Egyptian protests; and the 2012-13 Shahbag protests, i.e. one 'Storm' event and three 'Protest' events. We begin by examining the Hurricane Sandy event. To understand why this is a difficult event for the *xQuAS+Crowd* approach to summarize, we need to examine both the clusters that represent the possible information that should be included in the summary (see Section 6) and the crowd aspects that xQuAS used for diversification. Table 8 (a) and Table 8 (b) show the textual representations that the assessors provided for each information cluster for the Hurricane Sandy event and the crowd aspects produced for that same event, respectively. The important observation to make from Table 8 is that many of the information clusters extracted for Hurricane Sandy are very specific to that particular event. For instance, we have clusters representing the briefing of Barack Obama, the sinking of the tall ship Bounty and locations particular to this event, such as Staten Island. If we compare these cluster representations to the crowd aspects for the same event, it is clear that the broader crowd aspects may not well match the information clusters, resulting in updates containing relevant information to those clusters to be missed. Although it is out of the scope of this work, an interesting solution to this problem would be to pair the explicit approaches proposed here with event aspects identified in real-time as each event unfolds. For instance, one might use burst detection techniques or track trending phrases to find new relevant event aspects.

Table 8. Textual representations created by the assessors for each information cluster and the Crowdsourced event aspects used for summary diversification for event 6: Hurricane Sandy.

(a) Cluster Representations

Cluster No.	Cluster Description
1	Barack Obama briefing
2	outages power electricity Bernards Branchburg Watchung
3	New York City powerless and flooded
4	New York Marathon cancelled
5	Coast Guard allowed two ships back into New York Harbor
6	Laurence Harbor flooded
7	Obama well-received handling
8	Knocks Out 25 Percent of U.S. Cell Phone Towers
9	Love Works mission
10	Worcester County flooding damage
11	Dwyane Wade
12	Nor'easter
13	waiver fuel
14	surge winds Maryland
15	price gouging
16	damage unknown
17	American Red Cross
18	$ 30 to $ 50 billion
19	Consolidated Edison Company
20	Wells Fargo
21	federal disaster response
22	death toll rises to 90
23	Category 1 mph kph
24	Jamaica
25	watch National Hurricane Center NHC
26	moving centered
27	Staten Island
28	school
29	Hormel Foods
29	Food donations
30	Mitt Romney
31	restored power
32	Delaware damage
24	Jamaicans Jamaica
33	21 dead Caribbean
34	Frankenstorm tall
35	ship Bounty Captain
36	Carlsen park Hatteras
37	Island
38	$ 2 million donation

(b) Crowdsourced Aspects

Aspect A	Labels
flood alerts	flood alerts
	rivers flooding
	severe weather
	weather warning
injured people	injured
death	killed
	dead
	toll
damage	damage
	lines cut
	cost
homeless	homeless
services	lost
	affected
tracking	weather front
	moving
	forecast
aftermath	aftermath
alert	alert
	travel
	warning
storm	storm
	blizzard
	weather
	extreme
	hurricane
	snowfall
disruption	disruption
	flight
	transport
government	government
	response
	official
magnitude	magnitude
	category
state of emergency	state emergency

Indeed, new event aspects can be used within the explicit diversification approaches tested here, by dynamically adding new aspects to the aspect set $\mathcal{A}$ over time.

Next examining the remaining three events where the *xQuAS+Crowd* approach performed poorly upon (2011-13 Russian protests; 2012-13 Egyptian protests; and the 2012-13 Shahbag protests), we note that all of these belong to the 'Protest' type of event. Initially, we might think that the aspects for that event type produced via the crowd were poor for this event type. However, this is not the reason for the lower performance for these events. Rather, the source cause for the lower performance is that these (and other) protest events overlap in terms of timeframe, resulting in updates discussing the wrong event being selected. For instance, the update 'Cairo - An Egyptian opposition party on Monday claimed police tortured one of its members to death' was returned for event 23 'shahbag protests' when it was only relevant to event 20 'egyptian protests'. In general, 'protest' type events are difficult to summarize, as updates that from a language use perspective may seem to be relevant (because they contain protest-related keywords) are not. For instance, Table 9 illustrates five updates returned by the *xQuAS+Crowd* approach for event 20 'egyptian protests' which

Table 9. Examples of non-relevant updates returned by the xQuAS+Crowd system for Event 20: 'Egyptian Protests'.

Run	Timestamp	Update Text
xQuAS+Crowd	11/18/2012, 12:27am	Saturday , November 17 , 2012 President Barack Obama called Egyptian President Mohamed Morsy on Saturday to discuss the ongoing violence in Gaza.
xQuAS+Crowd	11/18/2012, 12:38am	Petraeus Testifies on Benghazi Attack Israeli Gaza strikes open gates of hell Worldwide protests against Gaza violence IAEA Iran ready to sharply increase nuclear work Barak approves increase in reservists call-up Iraq envoy backtracks on Arab action against Israel Comment Operation spreads its net More Stories
xQuAS+Crowd	11/18/2012, 12:32am	The Hamas website said Saturday that its leader, Khaled Meshaal, met with the head of Egyptian intelligence for two hours Saturday in Cairo, a day after the Egyptian official was in the Gaza Strip trying to work out an end to the escalation in violence.
xQuAS+Crowd	11/20/2012, 4:38am	Protests are being held around the world to support the end of the continuing violence.A Palestinian boy walks up the stairs of a house destroyed on Sunday by an Israeli strike in Gaza City, Monday, Nov. 19, 2012.
xQuAS+Crowd	11/29/2012, 11:35am	Thursday November 29 , 2012 , 3:19 am Reported incidents of sexual violence against women in the newest round of Egyptian protests are rampant across Twitter .

recieved a high relevancy score, but are not relevant. This highlights the need to better distinguish similar co-occuring events - particularly long running events like protest movements that are more likely to overlap.

8 CONCLUSIONS

In this article we proposed a novel approach to improve the performance of temporal summarization by explicitly diversifying the updates returned using aspects that are common to different event types. In particular, we proposed the use of explicit event aspects, representing the different types of information that a user might want to know about, as a means to guide sentence selection during summarization. We proposed two different methodologies to generate event aspects for an event type, based on Wikipedia infobox extraction and crowdsourcing respectively. We then extended a state-of-the-art explicit diversification framework to enable the use of those event aspects during the temporal summarization process. Through evaluation of temporal summaries generated for 46 events, we showed that diversifying summaries using explicit event aspects is overall more effective than classical novelty-based diversification, as the resulting summaries contain less off-topic and redundant content, while also being more timely. However, this can come at the cost of some summary comprehensiveness.

For future work, we aim to examine approaches to generate new event aspects in real-time as each event evolves. The proposed approach supports the dynamic addition of new aspects at any point. However, currently, event aspects are identified prior to the start of the event based on the event type, and hence only capture commonalities in information across events of the same type. Topical clustering techniques within each hour or across hours, such as those used by Li et al. [25] may help alleviate this issue. Nonetheless, these approaches will need to be extended to tackle scenarios where most of the input content is off-topic, as well as cases where topic-drift

occurs. Another interesting research direction is tackling value tracking, i.e. dealing with edge cases where updates are concerned with numerical statistics, such as the number of people killed, which fluctuate over time. The approach proposed in this work could be adapted to account for these special cases by dynamically revising the novelty of relevant event aspects when the most commonly stated values observed in the stream change.

Table 10. Run performances, pooling statistics and completeness for the TREC-TS 2013 top performing runs and the proposed systems from this article.

		Expected Latency Gain	Latency Comprehensiveness	Assessed @ 60	Returned @ 60
TREC	Top1	0.0794	0.1950	196	198
TREC	Top2	0.0675	0.2690	351	447
TREC	Top3	0.0500	0.2032	80	541
IASelect	Wikipedia	0	0	0	124
IASelect	Crowd	0	0	0	25
XQuAS	Wikipedia	0	0	1	133
XQuAS	Crowd	0	0	1	152

APPENDIX

A ILLUSTRATION OF COMPLETENESS ISSUES WITH THE OFFICIAL TREC-TS ASSESSMENTS

In Section 6 we produced an alternative set of ground-truth labels with which to test the proposed system. The reason for this is that the original ground-truth assessments pool provided by TREC only covers a small proportion of the sentences that a system might reasonably return. As such, it is quite probable that a new system will return relevant and informative updates that were not in the ground-truth assessment pool. Such a system would receive a low score, as non-assessed sentences are considered to be irrelevant.

This appendix includes an additional experiment to illustrate this issue and to show that it affects the explicit-diversification approaches proposed in this article. In particular, for the top three performing participating systems to the TREC 2013 Temporal Summarization track and for the four variants of the approach proposed in this article, we report in Table 10 their performances under the official TREC-TS 2013 evaluation metrics (Expected Latency Gain and Latency Comprehensiveness), the number of sentences that were returned for the 2013 topics (when considering the first 60 per-topic - 'Returned @ 60') and the number of those that were assessed by TREC assessors ('Assessed @ 60'). If a run produced by a system has a low 'Assessed @ 60' value then its reported performance is very likely to be misleading, as the metrics are assuming that the content it is returning is irrelevant in the absence of evidence.

From Table 10 we observe the following. First, the top performing system to TREC 2013 had almost perfect completeness, i.e. all of the sentences it returned (apart from 2) were assessed, hence its reported performance can be considered to be very accurate. Similarly, the majority (351/447) of sentences returned by the second best TREC system were assessed, meaning that we can have confidence that it's stated performance is accurate. In contrast, the third-best system that participated in TREC actually has a relatively low number of sentences assessed in contrast to the number returned (80/541). This means that there is a much higher chance that the performance of this run is under-estimated by the metric, as many of the sentences returned were not assessed and are assumed to be irrelevant.

More interestingly, examining the new approaches proposed in this article, we see that all of the four variants have a reported performance of 0. This is the result of the extreme case where none of the sentences returned by these approaches were originally assessed. It is for this reason that creating new ground-truth assessments is needed when testing new algorithms that are dissimilar to those in the original assessment pool.

B TREC TEMPORAL SUMMARIZATION METRICS

In this article we use the metrics developed for the TREC Temporal Summarization track to evaluate the performance of the temporal summaries produced. In this section, we provide the formulations of these metrics for reference. A more detailed explanation of these metrics can be found in [16].

The TREC Temporal Summarization metrics are designed to cover different aspects of evaluation: precision, comprehensiveness (recall), novelty, brevity and latency. Precise summaries are summaries that contain primarily relevant and interesting information. A summary that has a high comprehensiveness will include most of the information that a user might want to know about the event. A summary with good novelty will not return the same information to the user multiple times. A brief summary is one that is short and easy to follow. A low-latency summary will return new information to the user fast, i.e. as soon that that information becomes available. There are a variety of potential user models for temporal summarization, which describe what the user wants to know and how they consume the information. TREC Temporal Summarization assumes a 'push notification alert model'. In this case, only critical new information, certain to be of interest to most users, is expected to be returned as updates. The main properties that a summary should have under this user model are brevity, novelty and precision.

As it is difficult to capture all of the properties that we want to capture, two main metrics are defined to capture precision and comprehensiveness of a summary. The precision metric, referred to as *expected gain*, is the sum of the relevance of each nugget that an update is matched to. For a summarization system producing an update stream S, gain is computed as:

$$ExpectedGain(\mathrm{S}) = \frac{1}{|\mathrm{S}|} \sum_{u \in \mathrm{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u, n) \tag{5}$$

where $\mathbf{M}(u)$ is the set of gold standard 'nuggets' matching update u and $\mathbf{g}(u, n)$ measures the utility of matching update u with nugget n. For the purposes of the TREC track, these gold standard nuggets were manually extracted from the Wikipedia page for each event by human assessors, and represent the set of information that makes an update 'relevant' to an event.

On the other hand, the comprehensiveness metric, referred to as *comprehensiveness*, is the proportion of all nuggets matched by the system updates,

$$Comprehensiveness(\mathrm{S}) = \frac{1}{|\mathrm{N}|} \sum_{u \in \mathrm{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u, n) \tag{6}$$

where N is the set of nuggets for the current event.

Between them, these metrics capture precision, comprehensiveness and brevity. To provide a target metric, an F-like measure is also defined, referred to as *combined*, or H. This the harmonic mean of $\mathbf{G}$ and $\mathbf{C}$,

$$Combined(\mathrm{S}) = 2 * \frac{\mathbf{C}(\mathrm{S}) * \mathbf{G}(\mathrm{S})}{\mathbf{C}(\mathrm{S}) + \mathbf{G}(\mathrm{S})} \tag{7}$$

In order to reward novelty within a summary, a summary only receives gain the first time they return an update matching a nugget. Matches to updates later in the summary are ignored when computing Equations 5 and 6.

ACKNOWLEDGMENTS

This work is supported by the EC co-funded SUPER (FP7-606853) project.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. ACM, 5–14.

[2] James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 10–18.

[3] Omar Alonso and Ricardo Baeza-Yates. 2011. Design and implementation of relevance assessments using crowdsourcing. In *European Conference on Information Retrieval*. Springer, 153–164.

[4] Gaurav Baruah, Richard McCreadie, and Jimmy Lin. 2017. A Comparison of Nuggets and Clusters for Evaluating Timeline Summaries. In *Proceedings of the 26th ACM international conference on Information and knowledge management*.

[5] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia-A crystallization point for the Web of Data. *Web Semantics: science, services and agents on the world wide web* 7, 3 (2009), 154–165.

[6] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 875–883.

[7] 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*. ACM, 335–336.

[8] Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 815–824.

[9] Yun-Nung Chen, Yu Huang, Ching-Feng Yeh, and Lin-Shan Lee. 2011. Spoken Lecture Summarization by Random Walk over a Graph Constructed with Automatically Extracted Key Terms.. In *Interspeech*. 933–936.

[10] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. 2010. Overview of the TREC 2010 Web track. In *Proceedings of The 19th Text REtrieval Conference, TREC*, Vol. 10.

[11] John M Conroy, Judith D Schlesinger, and Jade Goldstein. 2005. Classy tasked based summarization: Back to basics. In *Document Understanding Conference*.

[12] Bruce Croft and John Lafferty. 2013. *Language modeling for information retrieval*. Vol. 13. Springer Science & Business Media.

[13] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, et al. 2014. Developing language processing components with gate version 8 (a user guide). *University of Sheffield, UK, Web: http://gate. ac. uk/sale/tao/index. html* (2014).

[14] Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of text analysis conference*. 1–16.

[15] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 65–74.

[16] Qi Guo, Fernando Diaz, and Elad Yom-Tov. 2013. Updating users about time critical events. In *European Conference on Information Retrieval*. Springer, 483–494.

[17] Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*. ACM, 517–526.

[18] Starr Roxanne Hiltz and Linda Plotnick. 2013. Dealing with information overload when using social media for emergency management: emerging solutions. In *Proceedings of the 10th international ISCRAM conference*. 823–827.

[19] Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K Usadi, and Xiaoyan Zhu. 2011. Generating breakpoint-based timeline overview for news topic retrospection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 260–269.

[20] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 63–72.

[21] Chris Kedzie, Fernando Diaz, and Kathleen McKeown. 2016. Real-Time Web Scale Event Summarization Using Sequential Decision Making. *arXiv preprint arXiv:1605.03664* (2016).

[22] Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting Salient Updates for Disaster Summarization.. In *ACL (1)*. 1608–1617.

[23] Jiwei Li and Claire Cardie. 2014. Timeline generation: Tracking individuals on twitter. In *Proceedings of the 23rd international conference on World wide web*. ACM, 643–652.

[24] Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. 2009. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international conference on World wide web*. ACM, 71–80.

[25] Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. 2011. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1137–1146.

[26] Chin-Yew Lin and Eduard Hovy. 2002. From single to multi-document summarization: A prototype system and its evaluation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 457–464.

[27] Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 510–520.

[28] Qian Liu, Yue Liu, Dayong Wu, and X ueqi Cheng. 2013. ICTNET at Temporal Summarization Track TREC 2013.. In *TREC*.

[29] Craig Macdonald, Richard McCreadie, Rodrygo L. T. Santos, and Iadh Ounis. 2012. From puppy to maturity: Experiences in developing terrier. *Open Source Information Retrieval* (2012).

[30] Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. Incremental update summarization: Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 301–310.

[31] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 573–580.

[32] Tu Ngoc Nguyen and Nattiya Kanhabua. 2014. Leveraging dynamic query subtopics for time-aware search result diversification. In *European Conference on Information Retrieval*. Springer, 222–234.

[33] Miles Osborne, Saša Petrovic, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2012. Bieber no more: First story detection using Twitter and Wikipedia. In *SIGIR 2012 Workshop on Time-aware Information Access*.

[34] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.

[35] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 881–890. https://doi.org/10.1145/1772690.1772780

[36] Rodrygo LT Santos, Iadh Ounis, and Craig Macdonald. 2015. Search result diversification. *Foundations and Trends in Information Retrieval* 9, 1 (2015), 1–90.

[37] Rodrygo Luis Teodoro Santos. 2013. *Explicit web search result diversification*. Ph.D. Dissertation. University of Glasgow.

[38] Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The Pythy summarization system: Microsoft research at DUC 2007. In *Proc. of DUC*, Vol. 2007.

[39] Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *European Conference on Information Retrieval*. Springer, 245–256.

[40] Tuan A. Tran, Claudia Niederee, Nattiya Kanhabua, Ujwal Gadiraju, and Avishek Anand. 2015. Balancing Novelty and Salience: Adaptive Learning to Rank Entities for Timeline Summarization of High-impact Events. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 1201–1210. https://doi.org/10.1145/2806416.2806486

[41] Maria Vargas-Vera and David Celjuska. 2004. Event recognition on news stories and semi-automatic population of an ontology. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 615–618.

[42] Dingding Wang and Tao Li. 2010. Document update summarization using incremental hierarchical clustering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 279–288.

[43] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 2008. Multi-document Summarization via Sentence-level Semantic Analysis and Symmetric Matrix Factorization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 307–314. https://doi.org/10.1145/1390334.1390387

[44] Jun Wang and Jianhan Zhu. 2009. Portfolio theory of information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 115–122.

[45] Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A Sentence Compression Based Framework to Query-Focused Multi-Document Summarization.. In *ACL (1)*. 1384–1394.

[46] Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. 2008. Query-sensitive Mutual Reinforcement Chain and Its Application in Query-oriented Multi-document Summarization. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 283–290. https://doi.org/10.1145/1390334.1390384

[47] Fei Wu and Daniel S Weld. 2010. Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 118–127.

[48] Tan Xu, Douglas W. Oard, and Paul McNamee. 2013. HLTCOE at TREC 2013: Temporal Summarization. In *Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013*. http://trec.nist.gov/pubs/trec22/papers/hltcoe-ts.pdf

[49] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 745–754.

[50] Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 334–342.

[51] Chunyun Zhang, Zhanyu Ma, Jiayue Zhang, Weiran Xu, and Jun Guo. 2015. A multi-level system for sequential update summarization. In *Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2015 11th International Conference on*. IEEE, 144–148.

[52] Guido Zuccon, Leif Azzopardi, Dell Zhang, and Jun Wang. 2012. Top-k retrieval using facility location analysis. In *European Conference on Information Retrieval*. Springer, 305–316.