Online News Tracking for Ad-Hoc Information Needs

Jeroen B. P. Vuurens
The Hague University of Applied Science
Delft University of Technology, The Netherlands
j.b.p.vuurens@tudelft.nl

Arjen P. de Vries
CWI
Delft University of Technology, The Netherlands
arjen@acm.org

Roi Blanco
Yahoo Labs, London, England UK
roi@yahoo-inc.com

Peter Mika
Yahoo Labs, London, England UK
pmika@yahoo-inc.com

ABSTRACT

Following online news about a specific event can be a difficult task as new information is often scattered across web pages. In such cases, an up-to-date summary of the event would help to inform users and allow them to navigate to articles that are likely to contain relevant and novel details. We propose a three-step approach to online news tracking for ad-hoc information needs. First, we continuously cluster the titles of all incoming news articles. Then, we select the clusters that best fit a user's ad-hoc information need and identify salient sentences. Finally, we select sentences for the summary based on novelty and relevance to the information seen, without requiring an a-priori model of events of interest. We evaluate this approach using the 2013 TREC Temporal Summarization test set and show that compared to existing systems our approach retrieves news facts with significantly higher F-measure and Latency-Discounted Expected Gain.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Clustering, Multi-document summarization

1. INTRODUCTION

Internet users are turning more frequently to online news as a replacement for traditional media sources such as newspapers or television shows. Still, discovering news events online and following them as they develop can be a difficult task. Although the Web offers a seemingly large and diverse set of information sources ranging from highly curated professional content to social media, in practice most sources base their stories on previously published works and add a much more limited set of new information. Thus users often end up spending significant amount of effort re-reading the same parts of a story before finding relevant and novel information. Most recently, the TREC Temporal Summarization track[1] have taken up this challenge, promoting research in the area of *online* news summarization, i.e. focusing on developing news, as opposed to archival news. Online summarization is a crucial aspect of real-world products such as online live streams for natural disasters, product launches, financial or political events, breaking news notifications on mobile devices and topical daily news summaries like Yahoo! news digest[2].

In this study, we propose a novel approach for the temporal summarization of news. Our approach works in an online fashion and provides previously unseen information related to a predefined ad-hoc information need, expressed as a user query. Contributions of this work are the use of a specifically designed clustering approach to detect news that is supported by multiple online providers, and the online selection of the best sentences according to a specifically tailored relevance model over recently seen information, that allows the retrieval of unanticipated information by adapting to information recently seen instead of requiring an a-priori model of events of interest, and requires no manual intervention and contains a small number of parameters that can be tuned in straightforward fashion.

We evaluate our approach using the 2013 TREC Temporal Summarization test set. In these experiments, our approach significantly outperformed the top performing systems on both F-measure and latency-discounted expected gain. To facilitate further research in this area, we also publish our implementation of the described model, the results of empirical experiments and the annotated ground truth[3].

The remainder of this paper is structured as follows: Section 2 discusses related work in the area of temporal summarization of online news information and the necessary prerequisites. In Section 3, we present our approach to extract sentences containing news facts from an online stream of news articles. In Section 4, we describe the implementation, test set used for the empirical evaluation, and how the data in the collection was processed. In Section 5, we present the results of the empirical evaluation, and analyze parameter sensitivity. The conclusions are presented in Section 6.

2. RELATED WORK

2.1 News tracking and summarization

The task of detecting events can be automated using information about the events published online. For this purpose, the Topic Detection and Tracking (TDT) program was initiated to discuss applications and techniques for detecting and tracking events that occur in real-time and the infrastructure to support common evaluations

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

[2] https://mobile.yahoo.com/newsdigest

[3] http://newstrackerpaper.github.io/

ICTIR '15 September 27-30, 2015, Northampton, MA, USA

DOI: http://dx.doi.org/10.1145/2808194.2809474.

of component technologies. The *tracking of news* involves the online identification of stories that discuss a targeted event, which needs to begin as soon as a only a few training documents have become available to model a real world setting. For this, Allan et al. present an information filtering approach, in which a tf-idf vector made from training documents is used as a query to match only documents that exceed a similarity threshold. In one experiment, "surprising" (previously rarely seen) words were used for tracking events, but they found that these words do not provide a broad enough coverage to capture all stories on the event and that many of these "surprising" words are useless for retrieval. They also found that a query based on initial training documents does not allow to track stories when the discussion of an event changes over time. For some queries at least, results were improved by using a tracking model that adapts the query based on new information seen, similar to the notion of pseudo-relevance feedback [5].

The temporal summary of news stories can help a person monitor changes in the coverage of news stories over time, which are typically very redundant and increase the effort required to identify genuinely new information [13]. The core technique of temporal summarization is to summarize multiple texts by extracting salient sentences. Regarding measures of salience that can be used to choose the best sentences for news summarization, the literature provides no clear consensus. Two general criteria to select the best candidates sentences are the most *useful* and *novel* sentences, i.e., related to the topic and non-redundant. Techniques that use these criteria for instance consider the words in the sentences, look for cue words and phrases, consider features such as sentence length and the case of words, or compare patterns of relationships between sentences. Often, these approaches use statistics from the corpus itself to decide on the importance of sentences, and some leverage existing training sets of summaries to learn the properties of a summary [3]. Candidate sentences can subsequently be ranked based on estimated importance, e.g. [11, 18, 21]. Some work has focused more specifically on the summarization of news in an online setting. Radev et al. presented a news delivery and summarization system "News In Essence", that supported retrieval of news related to a document that the user provided [19]. Gabrilovich et al. present a methodology for filtering news stories based on novelty, by selecting the articles that are most different to those already read [13]. This work also focuses on summarization in an online setting.

The salience of sentences is more easy to determine in retrospect than for online systems [23]. In retrospect, there is more information to compare possible solutions based on size and the coverage of the possible relevant facts over the stream of redundant information. Erkan et al. argue that sentences that are similar to many of the other sentences in a cluster are more central (or salient) to the topic, and propose an algorithm with resemblances the HITS algorithm that uses similarity edges instead of hyperlinks to estimate the salience of sentences [11]. Yu et al. present an approach to detect opinions in contrast to factual information with very high precision and recall, by using a fairly straightforward Bayesian classifier [24]. In recent work, Tran et al. used this classification approach in reverse to select headlines containing factual news. They further refined their detection of salient headlines by assuming that a higher *spread* indicates more important news, and that the relatedness to subsequent events indicates *influential* news, but it appears their approach is more specific for summarization in retrospect [21].

The clustering of information that discusses the same topic can be useful for several purposes related to temporal summarization. Some studies use the heuristic that the most similar sentences tend to be salient, which can be detected using clustering [12, 11, 14]. Clustering has also been used to extract concise information from redundant sources [4, 18]. Allan et al. experimented with different variants and obtained better results using single linkage clustering with cosine similarity [4]. In single linkage clustering, every data points is assigned to its nearest neighbor, in accordance to the k-Nearest Neighbor (kNN) decision rule described by Cover and Hart for classification [10]. They show that for the classification of n-samples there exists no $k \neq 1$ with a lower probability of error than $k = 1$ against all distributions. A known problem for finding nearest neighbors in large datasets is that the required number of computations increases quadratically [17].

The main contribution of this work is a novel approach to select the most salient sentences in a news stream, by leveraging the redundancy that is typical between news articles that discuss the same event. We introduce a variant of kNN clustering called 3-NN, which differs from existing work by forming clusters around a minimum of three sentences that are in each others' $k = 3$ sets of nearest neighbors and published by different news agents. For the online summarization of these salient sentences, we use an adaptive approach that resembles that of [5], but rather based on recently seen salient sentences to limit the selection of sentences to the most relevant according to the most recent news. The complete system we used to evaluate these efforts can be viewed as a hybrid combination of techniques for query based online news tracking and summarization, adapted from [5, 3]. Our approach differs by a stronger emphasis on novelty of information emitted (like [13]). Hereto, we estimate the amount of previously unseen information to use only sentences that are likely to contain novel information.

2.2 TREC Temporal Summarization

In recent years, TREC stimulated research on online summarization of news related to a specific topic or query, by initiating the Temporal Summarization track. The PRIS team participated with a manual system in the 2013 edition of the TS track and obtained the highest Expected Gain. They use hierarchical Latent Dirichlet Allocation on documents describing similar events as the topic to mine ten subtopic descriptions per TREC topic. From the generated topic descriptions they manually selected the keywords that describe each topic best. The sentences that are most similar to the selected keywords of a topic are selected as output [25]. In the same track edition, ICTNET obtained the highest F-measure of all participants. A list of relevant words is learned from training documents, which are then matched to the sentences of documents that contain all query terms in the title. A matching sentence is then compared to previously emitted sentences, and removed if the similarity exceeds a threshold [15]. These participants provided the best performing runs, out of 27 submitted for this task, and we will compare our results to the results of these systems in the evaluation. These query based approaches dominantly make use of a model crafted over similar events, e.g. other earthquakes or train crashes documented on Wikipedia. These approaches are optimized for retrieving the same, often reported types of information about common types of events, but may fail when the type of the event is not known or the type of information is not typical for the type of event. The proposed method uses only a single query to represent the event and does not require further training data.

3. DESIGN

News facts can be obtained from several sources on the Web, e.g. online news sites, blogs, social media, Wikipedia. One advantage over traditional broadcast news is that online news facilitates easy access to additional information. However, manually tracking relevant and novel news facts online is rendered inefficient by

the high redundancy between multiple sources that discuss more or less the same information. This research focuses therefore on the automated extraction of relevant and novel news facts for ad-hoc information needs, allowing to push newly published facts to a user the instant they are published; or, alternatively, to present the user a summary of the most important news facts over a timeline. Additionally, presenting the most important news facts on a timeline may also be useful to help keep update knowledge bases up-to-date, such as Wikipedia or the knowledge graphs used by search engine companies. From an end-user perspective, we consider it important that a high percentage of results is on-topic, and therefore this study uses news articles as the sole source. We expect their content to be mostly factually correct, timely, and presented in an accessible form [19]. Events that are of interest to many people are naturally reported in different news articles, from different sources [5, 9]. In our approach, we leverage the redundancy between news articles, clustering sentences that are likely to discuss the same news facts to select salient sentences and to avoid biased information [14]. Eventually, our work may serve as a baseline to evaluate approaches that also consider alternative sources like social media.

In this Section, we describe a new approach to extract sentences from an online stream of published news articles that are related to a user's ad-hoc query. We operate in a strict online setting, processing the articles one at a time as they arrive. The remainder of this Section first discusses observed characteristics for factual news. We outline the process that is proposed to extract sentences containing news facts from a stream of online news articles, followed by a detailed discussion of each step in this process.

3.1 News extraction process

We first outline the proposed method for the online tracking of ad-hoc user needs in a stream of news articles, which consists of three steps: *route*, *identify salient sentences* and *summarize*. The key method underpinning our approach is a clustering method that takes care of both the routing and the identification of salient sentences. In the first step, a single graph is maintained in which all news articles are clustered, and 'query matching clusters' are *routed* to a query specific module to identify salient sentences. In this second step, per query that is being tracked, we cluster the contents of clusters that match that query to *identify* the most central sentences, which we consider the most salient ones. In the third step, per query that is being tracked we *summarize* the salient information by qualifying only the most novel and useful sentences from the current document.

3.1.1 Routing

The first step of the outlined process identifies clusters of news articles by several news agents that share information, and route 'query matching clusters' to the designated identification and summarization process that is executed per query. Here, we define *query matching clusters* as the clusters that contain at least one news article that matches that query; in this study, an article matches a query when all query terms appear in its title. This section first gives a rationale for the features used to assign a document's nearest neighbors, and then describes the clustering method in detail.

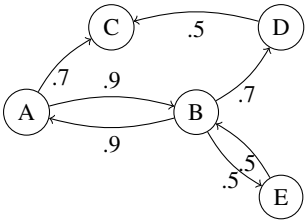
To estimate which news articles are likely to discuss the same event, we use the similarity of the titles and the proximity of the publication times. The use of titles is motivated by the observation of Tran et al., that news article titles are often short sentence abstracts of the news contained, to allow readers to gain a quick overview of the news based on titles and to invite them to read the full article if it is of interest to them [21]. Additionally, titles contain less words than entire documents, and so the collection of news article titles can be fitted into the memory of a single computer, allowing to process the data without the need to partition it. The latter is primarily a practical argument when developing an online news summarization approach. The use of proximity in publication times is motivated by the observation that stories about the same event often occur in proximate time, most particularly for unexpected events where the news media exhibit strong interest in a story [5, 23].

We introduce a *3-NN* streaming variant of k-Nearest Neighbor clustering, that assigns directed edges to each article's three nearest neighbors while not allowing nearest neighbor links within the same web domain. We use an online algorithm to detect newly formed clusters as 2-cores, according to the theory of k-degenerate graphs [16]. These 2-cores identify the most central information based on similarity in content, proximity in publication time and support by multiple news agents. The selected news is therefore is more likely to be factual, correct and important.

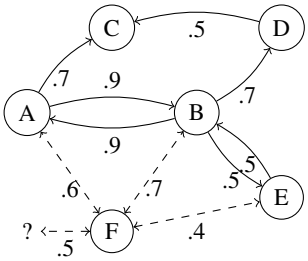
In 3-NN, a new 2-core is formed only when the arriving node is part of a bi-directional loop of nodes that is currently not clustered. Multiple bi-directional loops that are connected by a single bi-directional edge are considered to be separate clusters. Nodes that are not part of a 2-core are still assigned to a cluster if their majority of nearest neighbors is a member of the same cluster. Figure 1 illustrates the online process that takes place upon the arrival of new articles (that correspond to nodes in the graph), when clusters are formed, expanded or disbanded. Edges in the graph point to one of a node's k-nearest neighbors, labeled with the similarity between the nodes. Dashed arrows indicate the similarity between new arriving nodes and existing nodes.

Considering the example of Figure 1a in more detail, nodes A and B are not clustered because there is no evidence that the majority of both nodes nearest neighbors must belong to the same cluster (C and D could belong to different clusters). When a new node F arrives (Figure 1b), it is compared to the existing nodes, to assign its three nearest neighbors. Since F is more similar to B than B's currently weakest nearest neighbor E, an edge from B to F will replace the edge from B to E. After F has been added (Figure 1c), nodes A, B and F form a bi-directional loop. For this particular situation, we can deduce that A, B and F must have their majority of nearest neighbors in the same cluster, and therefore they form a cluster. We will refer to the nodes that form a bidirectional loop to establish a cluster as its *core nodes*. In Figure 1d, E is added to the cluster consisting of A, B, F, because its majority of nearest neighbors connect to that cluster. In Figure 1e, when a new node G arrives that is more similar to A than its weakest of nearest neighbors F is, the edge from A to F will be replaced by an edge from A to G. With this change, the bi-directional loop from which we deduced the existence of a cluster A, B, F, is now gone. Therefore, in Figure 1f, there is no more cluster.

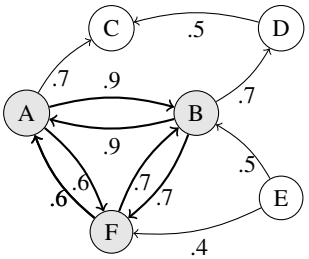
The nearest neighbors of a given title or sentence are found by computing the similarity to all other titles or sentences. The similarity between two sentences s_i and s_j is scored using Equation 1, which combines the cosine similarity between the binary vector representation of the two sentences with the difference in publication time, in accordance to the observations by [23]. Equation 2 estimates the temporal proximity of two publications, $\tau \in [0,1]$, as the absolute time between the publication times of $s_i.t$ and $s_j.t$, truncated by a constant maximum period T. Equation 3, δ is a function that guarantees that assigned nearest neighbors are published within a time span with duration T, and originate from a different source domain ($s_i.d \neq s_j.d$).



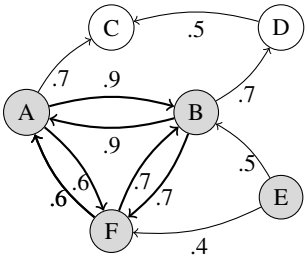
(a) The initial state has no clusters. Clusters are not formed on single connected subgraphs: C and D could have a majority of their nearest neighbors in two different clusters, which would lead to ambiguity in cluster assignment.



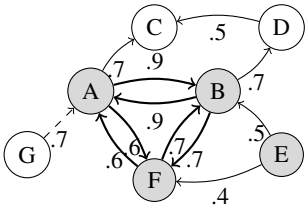
(b) When new node F arrives, edges of existing nodes to their weakest nearest neighbor are replaced if the new node is more similar. E.g., the edge from B to E is replaced by an edge from B to F.



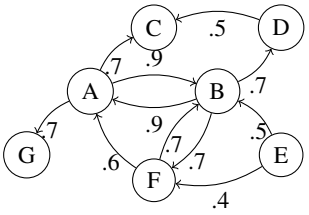
(c) A cluster is created when 3 or more nodes form an bi-directional loop. E.g., A, B and F form a cluster sharing the majority of their nearest neighbors.



(d) E is assigned to the same cluster, because the majority of its nearest neighbors lie in the cluster formed by A, B, F.



(e) Upon arrival of G, A loses its edge to F, breaking the bi-directional loop that justified assigning A, B and F to the same cluster.



(f) Consequently, nodes A, B, F, E no longer form a cluster. Note that the single connection from B and F to A is not sufficient to maintain the cluster, since A and D/E could be assigned to a different cluster each.

Figure 1: Explaining when clusters are created and broken using the nearest neighbor heuristic, $K = 3$, with the requirement that nodes are only clustered when they are members of a 2-degenerate core or when their majority of nearest neighbors is a member the same cluster.

$$sim(s_i, s_j) = cos\left(s_i, s_j\right) \cdot \tau\left(s_i, s_j\right) \cdot \delta\left(s_i, s_j\right) \tag{1}$$

$$\tau(s_i, s_j) = 1 - \frac{|s_i.t - s_j.t|}{T} \tag{2}$$

$$\delta(s_i, s_j) = \begin{cases} 0, & \text{if } |s_i.t - s_j.t| > T \text{ or } s_i.d = s_j.d \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

3.1.2 Identification of salient sentences

For each tracked query, we identify salient sentences in a separate graph. The routing will result in forwarding the clusters that match a query to the corresponding 'sentence graph', to which a node is added for every sentence in the query matching clusters. For sentences we follow an analogous rationale as for titles; salient sentences are likely to be published in proximate time and share information and are thus likely to be clustered together, and we therefore cluster the sentences according to the same 3-NN heuristics as described above. Within the clusters of such a 'sentence graph', the core nodes are the most central sentences and thus in this study regarded as the most salient. Operating in an online setting, we only consider sentences from the current document as *candidate sentences* for the news summary. However, if candidate sentences are clustered, their entire cluster will be passed to the summarization step, since the cluster provides part of the context needed to qualify (future) candidate sentences.

3.1.3 Summarization

In general, for an optimal summary of news we should select sentences that are the most useful and novel, i.e. related to the topic and non-redundant with other sentences in the summary [3]. In this step, we qualify which candidate sentence(s) are added to the news summary. Obviously, this is easier to optimize in retrospect than in an online setting, since we must decide whether or not to use a sentence without knowledge of what is yet to come. Operating in an online setting, we only consider sentences from the current document to use in the summary. Once the decision has been made to add a sentence to the summary, this cannot be reversed if the original sentence is removed from the cluster when a new sentence arrives. For the qualification we formulate a set of heuristics to select useful and novel sentences.

Erkan et al. hypothesize that sentences that are similar to many other sentences in the cluster are more salient to the topic [11]. In our 3-NN clustering, the core nodes are thus likely to be the most salient sentences. Initially, we expand clusters by adding non-core nodes that have a majority of nearest neighbors in one cluster. However, nodes can be assigned to a non-related cluster in the absence of closely related content. A directed path from a core node to another cluster member is likely to identify closely related content. To reduce the risk of using off-topic sentences, we apply a variant of graph peeling [1] by removing nodes to which there exists no directed path from a core node. In the remainder of this section when we refer to cluster members we only consider the cluster members for which a directed path exists from a core node.

In our approach, a redundant stream of news articles is aggregated into a concise summary by selecting only sentences that are most relevant to the most recent developments for the topic. Without the use of training documents, we obtain a model of the most important information from the news stream, however, what information is important for a topic can change over time [5]. Yang et al. observed that a time gap between bursts of topically similar stories is often an indication of different events, suggesting a need for

monitoring cluster evolution over time and a possible benefit from using a time window for event scoping [23]. If significant shifts in vocabulary indicate stories that report a novel event, this motivates the use of an adaptive model that allows to identify novel events. Analogous to [8], we propose an unsupervised 'berry-picking' approach that estimates relevance at some point in time based on the information seen in a window over the prior h hours, and compares the estimated relevance of the candidate sentences to sentences already summarized, to selectively qualify only candidate sentences that rank among the top-r sentences. The rationale for this berry-picking approach is that news topics tend to evolve over several subtopics; consider for example a crime happening, the police investigation, a suspect being arrested, etc. Some subtopics are repeatedly reported over a longer period, while others are mentioned only briefly. We construct a relevance model per news topic (a current 'event profile'), which is initially seeded with the user's query terms. The model is continuously expanded with the core node sentences from all query matching clusters to limit the risk of adding off-topic information. An adaptive relevance model is obtained at time t by removing sentences that were published before $t-h$ hours, allowing to shift the notion of relevance to recently seen information. In the event the relevance model contains no sentences published after $t-h$, the relevance model returns to the original query terms. For ranking, we express the relevance at a given a point in time as a word vector, where the frequency of each word is the number of sentences it appeared in over the last h hours. The candidate sentences of the latest arriving document are then ranked among the sentences currently in the summary, using the cosine similarity between each sentence and the relevance vector. Candidate sentences ranked outside the top-r are disqualified for use in the summary.

New sentences that share no words with information already seen can disorient the reader, being possibly off-topic as well. To reduce topical drift and improve readability of the timeline created, we require qualified sentences to contain at least one of the query terms and two words that appear jointly in either the query or in a sentence already used in the summary. Formally, in Equation 4, we define $WC(s)$ as the collection of all combinations of words (w_1, w_2) that appear in sentence s, and $QWC(s,q)$ as the subset of $WC(s)$ in which at least one of the words appears in the query q. In Equation 6, we define K as the collection of all word combinations containing at least one query term that was previously seen in either the query q or one of the sentences in the summary S. Finally, in Equation 7 is the constraint that at least on of the word combinations in the candidate sentence c must be in $K(S,Q)$. This simple requirement effectively filters out the (unrelated) sentences that still form clusters, such as navigational elements or links to other news stories.

$$WC(s) = \{(w_1, w_2) \mid w_1 \in s \wedge w_2 \in s \wedge w_1 < w_2\} \quad (4)$$

$$QWC(s,q) = \{(w_1, w_2) \in WC(s) \mid w_1 \in q \vee w_2 \in q\} \quad (5)$$

$$K(S,q) = \cup_{s \in S} QWC(s,q) \cup WC(q) \quad (6)$$

$$K(S,q) \cap WC(c) \neq \emptyset \quad (7)$$

Additionally, qualified sentences must add information that is not previously seen and is supported by another source. Previously unseen information could be simply measured by the number of previously unseen unigrams. Alternatively, the amount of information shared by sentences can be estimated by the number of two-word combinations that appear jointly in both sentences, which is possibly less affected by noise and will be used unless stated otherwise. Formally, in Equation 8, we define the set of possible sentences that can provide support for word combinations $SUP(CL,c)$, as the sentences s in cluster CL that are published on a news site $s.d$ that is different from the news site of the candidate sentence $c.d$. In Equation 9, we define the information gain G as the number of two-word combinations that appear in both the candidate sentence c and a sentence on a different news site, but not in one of the sentences that was used in summary S. In Equation 10, we set a threshold based on the number of possible word combinations that contains at least one non-query term. We use a parameter $g \in [0,1]$ to control the fraction of two-term combinations that must be gained to qualify a sentence to use in the summary.

$$SUP(CL,c) = \{s \in CL | s.d \neq c.d\} \quad (8)$$

$$G(c,CL,S) = |\cup_{s \in SUP(CL,c)} WC(s) \cap WC(c) - \cup_{s \in S} WC(s)| \quad (9)$$

$$G(c,CL,S) >= (|c-q|) \cdot (|c|-1) \cdot g \quad (10)$$

4. EXPERIMENT

4.1 Feasibility of online KNN Clustering

Clustering all news articles using the nearest neighbor heuristic, requires the computation of similarity of each news article against all others. For incremental online clustering, the number of required comparisons can be reduced by using a criterion to remove nodes and clusters that are outdated, by Aggerwal et al. referred to as 'cluster death' [2]. Since in this approach a zero score is assigned between sentences with a publication time more than T away (see Equation 3), we can do so with a high probability of not affecting clustering results. Since the news sentences of such a limited period of time fits into memory, we do not require an approximation such as Latent Semantic Hashing to partition the data. Additionally, we use in-memory posting lists on the words that appear in sentences, so that we do not compare sentences that have no word in common. In practice, this results in an algorithm of order $n \cdot log(n)$. Figure 2 shows the clustering efficiency over a stream of news articles in the KBA corpus, from 2011-11-06 until 2011-11-27, that was clustered on a standard laptop, in approx. 100 seconds. On the left-hand side of the graph, we observe that the clustering speed slows down slightly when more articles are in memory. The vertical drops in the graph are the result of removing 'expired' articles as discussed above. This graph shows online processing of all published news titles is feasible using the proposed clustering approach.

In the proposed 3-NN clustering method, nodes that do not have 2 nearest neighbors in the same cluster correspond to the outliers of Aggarwal and Philip [2] and remain un-clustered. In our experiments using sentences of news articles, on average 20% is unclustered at any given time.

In theory, a chain of nearest neighbors could span a period greater than T, and, although unlikely, cluster assignment could be affected over a larger time span. To allow for such anomalies, at the end of each day we prune sentences older than $T+1$ days, except for clustered sentences which are not pruned until all its members are older than $T+1$ days. For the 2013 KBA Streaming corpus, we compared the clustering results of a pruned run to a run that does not prune the articles from memory, and confirm that the clustering is not affected by removing 'expired' items.

4.2 Evaluation

To evaluate our approach, we used the test collection from the Sequential Update Summarization task at the 2013 TREC Temporal Summarization track, and compare effectiveness against the two best performing systems. For this track, the 2013 KBA Streaming

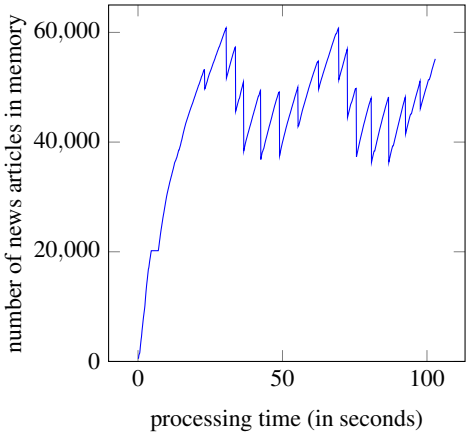


Figure 2: during the clustering of a stream of 3 weeks, the number of news articles in memory over time

corpus was used, in which the documents are already parsed into sentences by the organizers, and the sentence numbers are being referred to from the existing ground truth set. The task is to retrieve a list of timestamped extracted sentences (referred to as *updates*), for a set of 9 topics that contain a query referring to a news event. The effectiveness of a system is measured using a set of gold standard updates (referred to as *nuggets*), that were extracted from Wikipedia event pages and timestamped according to the revision history of the page. The TREC participants submitted a list of updates, from which a pool of 3,268 updates was manually compared to the 1275 identified nuggets for 9 topics, resulting in 2,416 matches between updates and nuggets and 2,142 updates that do not match a nugget (one update can match multiple nuggets).

Of the sentences returned in our experiments we found that an insufficient number has been annotated by TREC to obtain reliable metrics (see Section 5.4 for the empirical data and discussion). As a resolution, we manually annotated all missing sentences *against the existing nuggets*. During the annotation, to the best of our ability we retrieved similar results that were scored by TREC annotators to score our results consistently. Occasionally, we encountered updates that seemed very relevant but could not be matched to any nugget. Since we assume that no nuggets were added in the process of scoring the updates for participating TREC systems, we did not add any nuggets to the ground truth. Adding the updates to the pool without a matching nugget is equivalent to scoring these as irrelevant.

4.3 Data processing and cleaning

In an exploratory phase we used a crawl of online news articles over the first part of 2014 for construction and training of the system. For this crawl, we extracted a list of domains that are referenced on the Wikipedia Current Event Portal between January 1st 2013 and September 1st 2014, from the WikiTimes portal [20]. We removed all domains from Asia, Africa, non-English and non-news domains, resulting in 141 domains. For the evaluation on the KBA Streaming corpus, we use the same system and consider only news articles from the described domains.

The KBA Streaming corpus contains the original HTML source of the documents and sentences that were extracted by the organizers. This extraction was done using rudimentary heuristics, which in the absence of periods occasionally produced sentences of several hundreds of words that for instance include entire paragraphs, tables or navigational labels. Since our approach specifically depends on the quality of title clusters, we extracted the actual document titles from within the HTML title tags, stripped non-news elements (e.g. categories and news paper names) using a manually constructed list of general and domain specific regular expressions (e.g. truncating titles after the a dash and removing the word TIME if this was the last word in a title from the `time.com` domain). These actual titles are used for clustering the articles. For a fair comparison of the proposed model to the best TREC participants, we performed a *no titles* run that emits only sentences as extracted by the TREC organizers and thus is conform to the TREC guidelines. In Section 5.1, we will compare the performance to a run that does allow *HTML titles* to be emitted.

For processing, all sentences were tokenized by separating tokens on non-alphanumeric characters, the tokens were lowercased, and stop words were removed, however, we did not use any stemming.

4.4 Parameter settings

The approach proposed in this paper contains several parameters: k as the number of nearest neighbors used for kNN clustering, T as a time period used to discount the difference in publication time in the similarity function (Equation 2), r for the rank to be obtained to qualify a sentence to use in the summary (Section 3.1.3), l for the maximum length allowed for sentences used (Equation 3.1.3), h for the time in hours used for the relevance model (Section 3.1.3), and g to control the minimum amount of new information an qualified sentence must have (Equation 10). In the exploration phase of this research we analyzed the effect of these parameters on seven topics that were annotated using the guidelines of the TREC TS track, and on online news that is tracked in a live demo [22].

For the number of nearest neighbors, we used a fixed setting $k = 3$. By using an odd number of nearest neighbors there is no need to resolve ties. A value of $k > 1$ increases the likelihood to cluster around information that is supported by several news domains, while compared to high settings for k a low setting for k is likely to retrieve news faster and may improve recall. We leave the comparison of different values of k for future work, noting that this may be especially useful in more redundant domains like social media. For T, we have used a fixed setting $T = 3$ days throughout our study, based on the observation that it is not uncommon for news providers to post news that is more than a day old and allowing these articles to be clustered with the same content brought more promptly by other providers. Each of the remaining parameters was added to restrain the model in some respect. We observe that clustering results may vary largely dependent on parameter settings, which is possibly due to the high redundancy that is typical for news collections. The necessity of new manual annotation for each clustering outcome renders parameter training practically infeasible. Despite the variation in clustering results, the overall system performance is largely unaffected by changes in parameters. Therefore, we use a set of default settings $r = 5$, $l = 20$, $h = 1$ and $g = 0.3$, and will show in Section 5.2 that the model performance is insensitive to parameter sweeps.

5. RESULTS

5.1 Comparison of temporal summarization

For evaluation of the proposed method, we follow the guidelines of the Sequential Update Summarization task of the 2013 TREC Temporal Summarization track [6]. The effectiveness is measured using *Mean Expected Gain*, and *Mean Comprehensiveness*, which are similar to the traditional notions of respectively precision and recall in information retrieval systems, and we additionally use the

Mean Latency Discounted Expected Gain in which the gain is discounted based on the difference between the time of the first update that matches a nugget and the time the corresponding fact was added to Wikipedia. Formally, in Equation 11, the gain G of an update u in a set of updates S is based on a gain function g on the nuggets n for which u is the earliest matching update as returned by the function M^{-1}. In Equation 13, the Mean Expected Gain MEG_v for a system is the average gain over a set of events ε, for each of which the system produced sets of updates S^e (emitted sentences), for g (Equation 11) a binary function is used that returns 1 if an update matches a nugget, and the total gain is normalized by the verbosity of the updates $V(u)$ (Equation 12), which discounts by the number of words in u that are not part of an earliest matching string for a nugget divided by the average number of words in the strings of nuggets $|words_n|$. In Equation14, the Comprehensiveness C for a set of updates S for a specific event is number of matched nuggets G divided by the number of available nuggets for the event $|N|$. In Equation 15, the Mean Comprehensiveness MC is computed over all events ε. The Latency-Discounted Expected Gain is a variant of Equation 13 by using a modified function g (Equation 11) in which the binary relevance of matched nuggets is discounted by a monotonicaly decreasing function over the difference between the time of the earliest matching update en the time it was put on Wikipedia. For more details regarding these metrics, we refer to [6]. We also report the variant of the *F-measure* that summarizes the Expected Gain and Comprehensiveness in one metric and was used as the primary metric of the 2014 Temporal Summarization track.

$$G(u,S) = \sum_{n \in M^{-1}(u,S)} g(u,n) \tag{11}$$

$$V(u) = 1 + \frac{|all\_words_u| - |nugget\_matching\_words_u|}{avg|words_n|} \tag{12}$$

$$MEG_v = \frac{1}{|\varepsilon|} \sum_{e \in \varepsilon} \left(\frac{1}{\sum_{u \in S^e} V(u)} \sum_{u \in S^e} G(u,S^e) \right) \tag{13}$$

$$C(S) = \frac{1}{|N|} \sum_{u \in S} G(u,S) \tag{14}$$

$$MC = \frac{1}{|\varepsilon|} \sum_{e \in \varepsilon} C(S^e) \tag{15}$$

In Table 1, we compare four variants of our approach with the top participants of the Temporal Summarization Track. The "no titles" variant only uses the extracted HTML titles for clustering, but never uses these in the summary, therefore the results of this run are conform the track guidelines and comparable to other TREC participants. For the "HTML title" variant, we additionally allowed emission of the actual HTML titles which was not an option for TREC participants, the "unigram" variant is the same as "HTML titles" except that it measures new and previously seen information using unigrams instead of two-word combinations, and the "IDF weighted" variant uses the inverse document frequency obtained from Wikipedia on January 2012 (which predates the test collection) to compute the cosine similarity between sentences.

The results show that the "no titles" variant is significantly more effective than the top TREC in both F-measure and Latency-Discounted Expected Gain. Statistical significance was tested using a paired Student t-Test, 2-tailed, $p < 0.05$. Given the low number of topics, we also tested significance using Wilcoxon Signed-Rank test, 2-tailed, $p < 0.05$, which confirmed the significant improvements for all but the improvement in F-Measure of the "unigram" variant over ICTNET. At the topic level, our approach was outperformed by PRIS on topic 5 "Hurricane Isaac", and by ICTNET on topic 6 "Hurricane Sandy", using approaches that target words typically seen on the Wikipedia pages of hurricanes such as wind speeds, casualties and damage.

Table 1: Comparison of performance using the 2013 TREC TS track against the top participants. † significant improvements over PRIS, ‡ significant improvement over ICTNET, using paired Student t-Test, 2-tailed, $p < 0.05$

System	Expected Gain	Latency DEG	Comprehension	F
PRIS-cluster5	0.1491	0.1364	0.0994	0.060
ICTNET-run2	0.1024	0.1270	0.1921	0.067
no titles	0.2607 ‡	0.3067 †‡	0.1778	0.106 †
HTML titles	0.2449 ‡	0.3019 †‡	0.1901	0.107 †‡
unigram	0.2474 ‡	0.2934 †‡	0.1700	0.101 †‡
IDF weighted	0.2100 ‡	0.2763 †‡	0.1664	0.093 †

Compared to the "no titles" variant, the "HTML titles" variant obtains higher Comprehensiveness and a relatively higher Latence-Discounted Expected Gain. Possibly, some facts are only used in titles and some facts are introduced in titles before they are used in sentences. In Section 5.3, we look into the differences observed between these variants in more detail.

5.2 Parameter sensitivity

To study sensitivity of the effectiveness to variation of parameters, we performed parameter sweeps for the "HTML titles" variant, and plotted the results in Figure 3. During each sweep we changed only one parameter, using the default settings described in Section 4.4 for the remaining parameters.

Interestingly, we observe that the efficiency is insensitive to the size h of the time window used to estimate a relevance model of recently seen information. Possibly, if news is important it is mostly reported by different agents within half an hour, explaining why the effectiveness is comparable for a window of that size. For the rank r a sentence must obtain to qualify, we expected an increase in Comprehensiveness when increasing the size, but this effect is only observed for $r < 5$. For g, which controls the minimum amount of new information a qualified sentence must add to the summary, a low g will more greedily use sentences with a relative small amount of new information, resulting in a classic trade-off of recall for precision. In these experiments, setting $g > 0.5$ hurts performance, possibly because sentences that contain novel information often include previously seen information.

On this particular test collection, sentence extraction by the TREC organizers occasionally resulted in large parts of content being mistaken for a sentence, to which our model is particularly sensitive. When our approach is used on a stream of correctly parsed news sentences, the maximum sentence length l could become obsolete, since the results show a higher setting of l results in slightly higher comprehensiveness and F-measure. However, these metrics do not take into account that shorter sentences improve readability, which may be preferable on mobile platforms.

In our evaluation, the difference in performance for alternate parameter settings is marginal when compared to the difference with competing systems. Therefore our default parameter settings are not likely to overfit the model to the data.

5.3 Model variants

In Figure 4, we compare the performance between the four variants of the proposed model in more detail, by changing the mini-

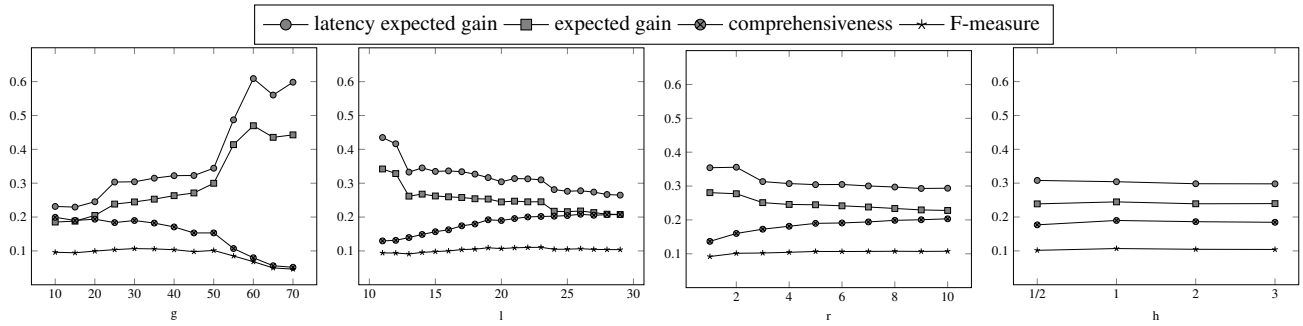


Figure 3: Impact of parameter values in the model performance, g=percentage of new word-pairs, not yet in the summary and co-occurring in the cluster, l=the maximum number of unique non stop words in a sentence, r=the minimum relevance rank amongst output sentences, h=number of (past) hours used to estimate the relevance model

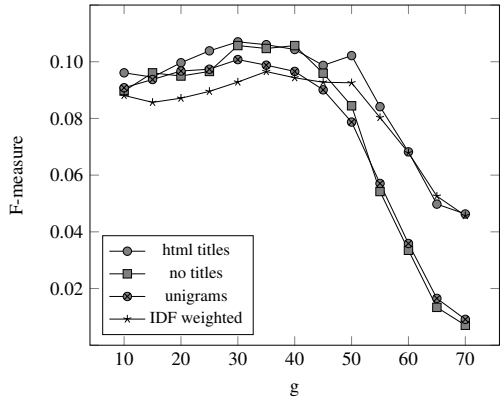


Figure 4: Comparison of the F-measure of model variants over different minimal information gains (g), a variant that is allowed to emit html titles, a variant that does not emit titles, a variant that estimates information gain using unigrams instead of word combinations, and a variant that uses IDF to estimate the cosine similarity for clustering sentences.

mum amount of new information g required to qualify sentences. For $g < .5$, the "no titles" variant obtains results that are very close to the "HTML titles" variant, indicating that most nuggets are also found in non-title sentences of the redundant news stream. The described approach does not use term weighting, except for the variant named "IDF weighted". Our experiments show that using IDF for the estimation of similarity between news sentences hurts effectiveness. One observation is that relevant news sentences often contain numbers, however especially low numbers have relatively low IDF weights in most collections. Lastly, measuring the amount of previously unseen information using unigrams is less effective than using 2-word combinations.

The analysis of different variants shows that all variants outperform the existing systems for $g < 0.5$, indicating that using 3-NN clustering of sentences combined with the qualification of sentences against a relevance model over recently seen information does improve over current state-of-the-art approaches.

5.4 Groundtruth

According to the TREC definition, for the computation of Expected Gain, non-annotated sentences are ignored. For this study, only 5 of the 529 sentences in our main run had been annotated, which is clearly insufficient for a reliable estimation for both Expected Gain and Comprehensiveness, as can be seen in Table 2. Baruah et al. found that duplicate sentences in the KBA corpus have not been added to the official TREC ground truth [7]. Therefore, for a system that returned a sentence that was annotated, results were different than for a system that returned an un-annotated duplicate of that same sentence. They extended the official ground truth with duplicate sentences in the collection, which we labeled the "Waterloo extended" set in Table 1. This extended ground truth set contains 38 of the sentences we returned. However, the results show an overestimation of Expected Gain, presumably because between systems there is more likely an overlap in relevant sentences than there is in non-relevant sentences. According to our observation, neither the official TREC ground truth nor the Waterloo extended set suffice for the evaluation of an external system; sentences missing in the existing ground truth would have to be annotated.

Table 2: Comparison of the performance of our HTML titles run, over the official TREC ground truth, the Waterloo extended set and a fully annotated set.

Ground truth set	Expected Gain	Latency DEG	Comprehension	F
TREC official	0.3224	0.4337	0.0149	0.014
Waterloo extended	0.2741	0.3640	0.0356	0.032
Fully annotated	0.2449	0.3019	0.1901	0.107

5.5 Example of cluster in action

In Figure 5, we show a real example how a news article from the KBA corpus was processed for topic 4 "sikh temple shooting", from an article from which 2 sentences qualify for emission using the default ranking requirement $r = 5$. At 6:28pm a new article arrives, for which a node is added to the the title clustering. The nearest neighbors for the nodes are updated, and the new node forms a bi-directional loop with two of its nearest neighbors, thus a new cluster is formed. At least one of the cluster members contains all query terms in its title, therefore all articles in the 'query matching cluster' are routed to the sentence clustering graph for that query. To this graph, all sentences in the articles of the 'query matching cluster' are added, but only the article of the current document (6:28pm) are candidates to be added to the summary. Two candidate sentences become a member of a sentence cluster and therefore these are checked if they qualify. First, the Relevance model is updated by removing outdated sentences and adding the core node sentences that are not a candidate sentence. Then the candidate sentences are ranked in a list with the sentences already

in the summary using to the relevance model. In this example, both sentences satisfy the requirements for novelty, old information and rank in the top-5. The qualified sentences are added to the summary and the candidate sentences are added to the Relevance Model.

6. CONCLUSION

In this study, we propose an approach for online temporal summarization of news related to ad-hoc information needs, expressed as a user query. In this approach, sentences are clustered based on cosine similarity, proximity in publication time and being supported by different news providers. The news extraction proceeds in three phases, first the titles of all incoming news articles are clustered, then we select the clusters in which the query terms appear and cluster the sentences contained in the clustered articles, and finally qualify a sentence as output when it contains sufficient new information and is more relevant than the top sentences already in the summary. Our approach requires no a-priori model that separates news containing sentences from other content for an event type or in general, and can therefore be used to extract relevant news facts without knowledge about the type of the news event, and requires no manual intervention and contains a small number of parameters that can be tuned in straightforward fashion.

We evaluated the performance against the best systems using the 2013 TREC Temporal Summarization track test set. Our approach significantly improved results over the existing systems in F-measure and Latency-Discounted Expected Gain. Results indicate that news on average is reported before it was added to Wikipedia. Since in the crawled collection the publication time was estimated to be the crawl time, it is reasonable to expect further improvement in latency for a system that monitors news sites in real-time for new publications.

We explain the effectiveness of the approach by our focus on information that is support by several news providers and has a strong relatedness to the original query. However, as described, this approach is also likely to have limitations regarding the recall that can be obtained. Specifically, the requirement that a cluster contains a sentence that contains all words in the query makes the method more suitable to minimal queries than for elaborate queries or when the topic is likely to described using synonyms. An interesting direction for future work is to study how these constraints may be alleviated to improve recall.

References

[1] J. Abello and F. Queyroi. Fixed points of graph peeling. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 256–263. IEEE, 2013.

[2] C. C. Aggarwal and S. Y. Philip. On clustering massive text and categorical data streams. *Knowledge and information systems*, 24(2):171–196, 2010.

[3] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18. ACM, 2001.

[4] J. Allan, V. Lavrenko, D. Malin, and R. Swan. Detections, bounds, and timelines: Umass and TDT-3. In *Proceedings of Topic Detection and Tracking Workshop (TDT-3)*, pages 167–174. Vienna, VA, 2000.

[5] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM, 1998.

[6] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, and T. Sakai. TREC 2013 Temporal Summarization. In *Proceedings of the 22nd Text Retrieval Conference (TREC), November*, 2013.

[7] G. Baruah, A. Roegiest, and M. D. Smucker. The Effect of Expanding Relevance Judgements with Duplicates. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014.

[8] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.

[9] H. L. Chieu and Y. K. Lee. Query based event extraction along a timeline. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–432. ACM, 2004.

[10] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[11] G. Erkan and D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)*, 22(1):457–479, 2004.

[12] J. G. Fiscus and G. R. Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking*, pages 17–31. Springer, 2002.

[13] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th international conference on World Wide Web*, pages 482–490. ACM, 2004.

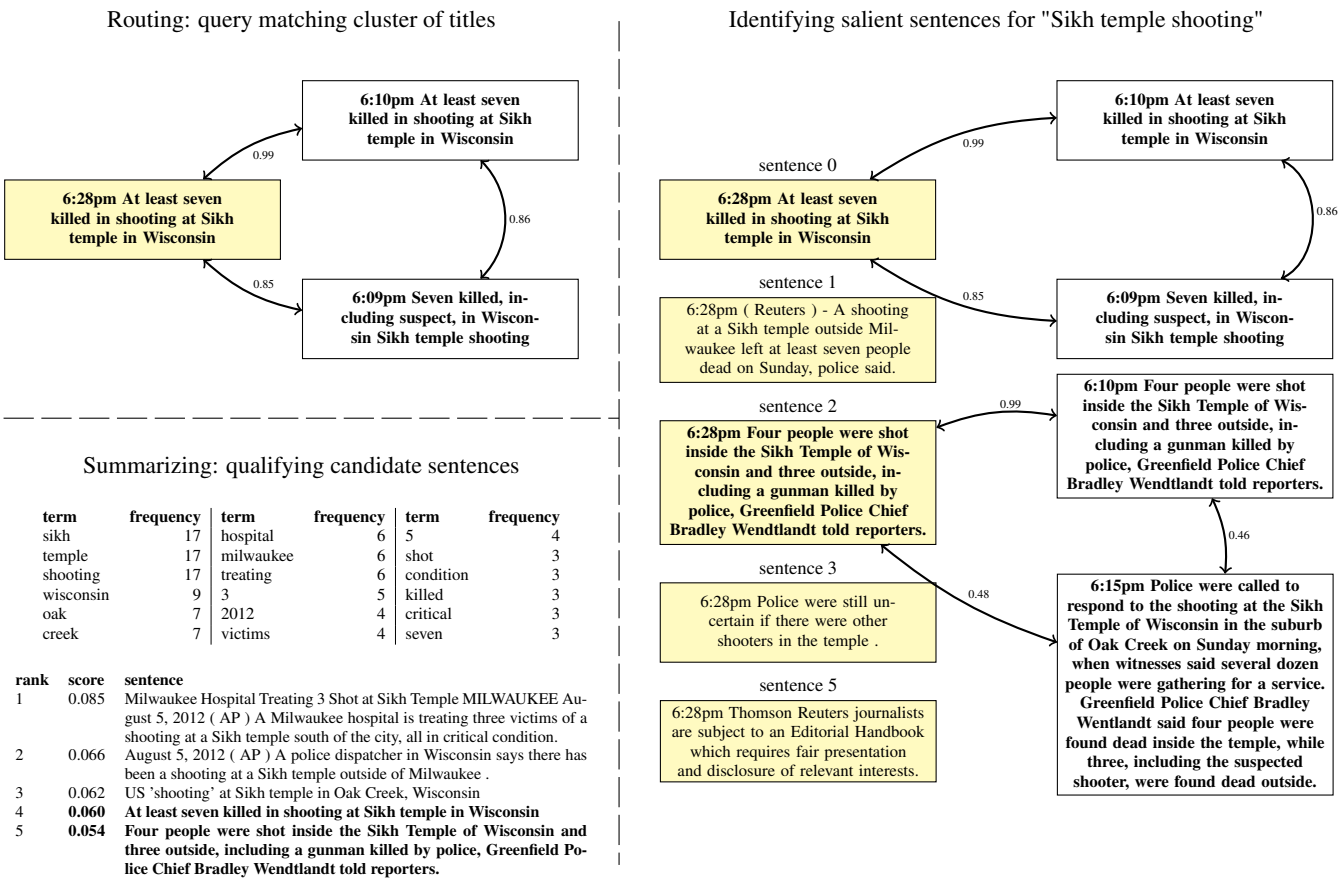
[14] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event registry: learning about world events from news. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 107–110. International World Wide Web Conferences Steering Committee, 2014.

[15] Q. Liu, Y. Liu, D. Wu, and X. Cheng. ICTNET at temporal summarization track trec 2013. In *Proceedings of the The Twenty-Second Text REtrieval Conference*, 2013.

[16] P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, and M. Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[17] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[18] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, et al. MEAD-a platform for multidocument multilingual text summarization. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 2004.



term	frequency	term	frequency	term	frequency
sikh	17	hospital	6	5	4
temple	17	milwaukee	6	shot	3
shooting	17	treating	6	condition	3
wisconsin	9	3	5	killed	3
oak	7	2012	4	critical	3
creek	7	victims	4	seven	3

rank	score	sentence
1	0.085	Milwaukee Hospital Treating 3 Shot at Sikh Temple MILWAUKEE August 5, 2012 (AP) A Milwaukee hospital is treating three victims of a shooting at a Sikh temple south of the city, all in critical condition.
2	0.066	August 5, 2012 (AP) A police dispatcher in Wisconsin says there has been a shooting at a Sikh temple outside of Milwaukee .
3	0.062	US 'shooting' at Sikh temple in Oak Creek, Wisconsin
4	**0.060**	**At least seven killed in shooting at Sikh temple in Wisconsin**
5	**0.054**	**Four people were shot inside the Sikh Temple of Wisconsin and three outside, including a gunman killed by police, Greenfield Police Chief Bradley Wendtlandt told reporters.**

Figure 5: A concrete example to illustrate data processing. Routing. A new article arrives at 6:28pm, and its title is added to a nearest neighbor graph of all existing titles. After assigning its nearest neighbors, it is part of a query matching (title) cluster, and therefore is routed to identify salient sentences. Identification. All sentences in the query matching cluster are added to the query's sentence clustering graph, the sentences from the current document being candidate sentences for the summarization. Two candidate sentences are clustered in sentence clusters that match the query and thus forwarded to the summarization of news for that query. Summarize. The core node sentences from the query matching sentences are added to the Relevance Model. The candidate sentences are ranked with the sentences that were already used in the summary. Both candidate sentences qualify because they are comprehensible (limited length and containing old information), contain a sufficient amount of new information and rank in the top-5. The two qualified sentences are added to the summary and the candidate sentences are added to the Relevance Model.

[19] D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. NewsInEssence: summarizing online news topics. *Communications of the ACM*, 48(10):95–98, 2005.

[20] G. B. Tran and M. Alrifai. Indexing and analyzing Wikipedia's current events portal, the daily news summaries by the crowd. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 511–516. International World Wide Web Conferences Steering Committee, 2014.

[21] G. B. Tran, M. Alrifai, and E. Herder. Timeline summarization from relevant headlines. In *Proceedings of the IR research, 37th European conference on Advances in information retrieval*, 2015.

[22] J. B. P. Vuurens, A. P. de Vries, R. Blanco, and P. Mika. Online news tracking for ad-hoc queries. In *SIGIR Demo*. ACM, 2015.

[23] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

[24] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics, 2003.

[25] C. Zhang, W. Xu, F. Meng, H. li, T. Wong, and L. Xu. The Information Extraction systems of PRIS at Temporal Summarization Track. In *Proceedings of the The Twenty-Second Text REtrieval Conference*, 2013.