CWI and TU Delft at the TREC 2015 Temporal Summarization Track

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

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. In previous work, we presented a novel approach to extract sentences from an online stream of news articles that summarizes the most important news facts for a given ad-hoc information need, which compared to existing systems obtained relatively high-precision results and a comparable recall [9]. In this track, we experiment with this approach to improve the recall of retrieved results.

2. DESIGN

2.1 News extraction process

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 [1] .

This track submission is based on previous work [9], in which we propose to extract the most salient sentences from an online news stream using a three-step process: *route*, *identify salient sentences* and *summarize*. The key method underpinning this approach is a clustering method that takes care of both the routing and the identification of salient sentences. In the first step, the news articles are added in timely order to a clustering graph that aims to cluster news articles that discuss the same news, and identifying cluster that contain a news article that matching the topic as a ‘topic matching cluster’ . In the second step, per topic, we cluster the contents of clusters that match the topic to *identify* the most central sentences, which we consider the most salient ones. In the third step, we *summarize* the salient information by qualifying only the most novel and useful sentences from the current document.

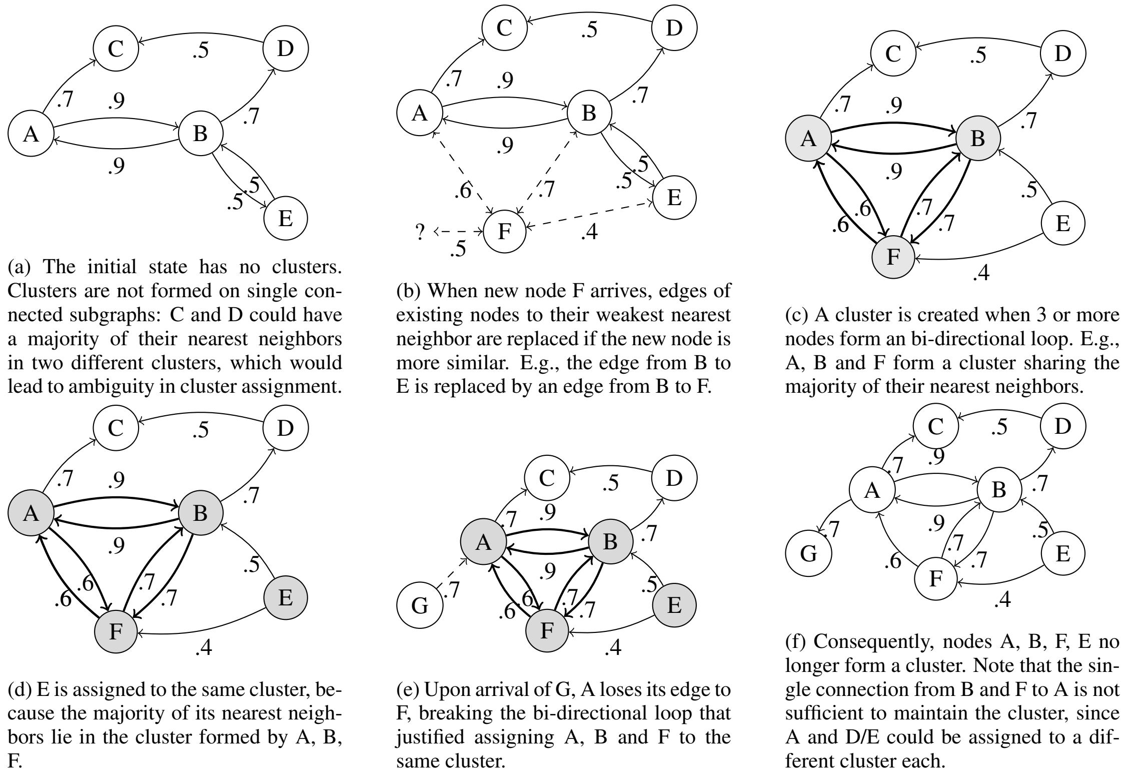
2.2 3-Nearest-Neighbor clustering

For clustering of information, we use 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 [9]. In this clustering graph we detect newly formed clusters as 2-degenerate cores, according to the theory of k-degenerate graphs [5]. These 2-degenerate 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 Figure Figure 1, we illustrate 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 using 3-NN. 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. For the forming of clusters, we consider only bi-directional edges and form a 2-degenerate core when the arriving node and a group of previously unclustered nodes are all connected to at least 2 other nodes in the newly formed cluster. The most common scenario is a triangle of 3 nodes, but larger cores do occur. Nodes that are not part of a 2-degenerate core can still assigned to a cluster, during step 1 if their majority of nearest neighbors is a member of the same cluster, and in step 2 if there additionally exists a directed path to that node from one of the core nodes. For more information on 3-NN clustering, we refer to [9].

2.3 Selection of top ranked sentences

Following [9], 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 [2]. 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 [10]. 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 [3], 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 compare 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 for a news topic (a current ‘event profile’), which is initially seeded with the terms that appear in the topic’s query. The model is continuously expanded with the core node sentences from all topic 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



(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.

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.

2.4 Normalized Information Gain

In the original work [9], cosine similarity was used to measure the similarity between sentences. In a previous study, for the task of constructing a hierarchical clustering of the sub news stories that are contained in the retrieved summary for a given topic, we proposed to cluster news articles that are likely to discuss the same news story using normalized information gain [7]. In that study, the use of normalized information gain appeared far more effective than using cosine similarity.

Following [7], the dissimilarity (or impurity) between documents is estimated by a normalized version of the *information gain*. Information gain has been successfully used when considering a fixed number of non-sparse dimensions of a data set [6, 4], however, in text collections the information gain is not comparable between subsets that use a different number of features. We introduce *normalized information gain* to address this problem, a measure that returns 0 for identical subsets and 1 for disjoint subsets. Formally, in Equation 1 H is the entropy over the words w in a bag of words s, given the size of the content c and $f_{w,s}$ is the frequency of word w in s. In Equation 2, the information gain IG is defined for separating a group of content $s+t$ into two separate bags of words s and t, with $|s|$, $|t|$ and $|s+t|$ as the number of words contained. In Equation 3, IG_{max} is the maximum information gain that would be obtained given the sizes of data subsets t and s if these are completely disjoint, and in Equation 4 IG is divided by IG_{max} to normalize IG_{norm} to a value in $[0,1]$. We converted this dissimilarity measure into a similarity measure by simply using $1 - IG_{norm}$. To compare the difference between these cosine similarity and normalized information gain, the run titled "docs" clusters news articles in step 1 using cosine similarity between the entire documents, and the run titled "IGn" alternatively uses normalized information gain.

$$H(s,c) = -\sum_{w \in s} \frac{f_{w,s}}{c} log_2 \frac{f_{w,s}}{c} \tag{1}$$

$$\begin{aligned} IG(s,t) =& H(s+t, |s|+|t|) \\ & - \frac{|s|}{|s|+|t|} \cdot H(s,|s|) - \frac{|t|}{|s|+|t|} \cdot H(t,|t|) \end{aligned} \tag{2}$$

$$\begin{aligned} IG_{max}(s,t) =& H(s, |s|+|t|) + H(t, |s|+|t|) \\ & - \frac{|s|}{|s|+|t|} \cdot H(s,|s|) - \frac{|t|}{|s|+|t|} \cdot H(t,|t|) \end{aligned} \tag{3}$$

$$IG_{norm}(s,t) = \frac{IG(s,t)}{IG_{max}(s,t)} \tag{4}$$

2.5 Improving Precision/Recall

We experimented with the above approach in a live demo [8], and using the TREC 2014 Temporal Summarization track as training set. During these experiments we observed that the above approach works better for under-specified queries than for over-specified queries, e.g. "Buenos Aires train accident" as an over-specified query is likely to miss news articles that do not contain all words, while the

Table 1: Results for our runs on the 2015 TREC TS track and the average over all TREC participants for task 1. * indicates runs that were not included in the annotation pool, but results were estimated based on the sentences that were annotated.

System	expected gain	latency expected gain	comprehensiveness	latency comprehensiveness	F(leg, lc)
titles	0.192	0.101	0.311	0.211	0.115
IGn	0.162	0.091	0.514	0.344	0.125
IGnPrecision	0.189	0.108	0.468	0.300	0.140
IGnRecall*	0.161	0.088	0.509	0.326	0.122
docs	0.124	0.085	0.468	0.342	0.122
docsRecall*	0.122	0.082	0.464	0.324	0.118
TREC-avg	0.158	0.098	0.447	0.320	0.131

under-specified queries "Buenos Aires train" or "Buenos Aires" for large crisis are likely to be dominated by the train accident in the given time interval and therefore the higher recall is often not at the expense of (much) lower precision. The original approach selects only cluster that contain a news article with all the query terms in its title, which for over-specified queries can result in a very low recall. Alternatively, all runs with a name other than “titles" match news articles that contain all query terms in the entire document rather than only the title.

To analyze the effect of changing parameters on the tradeoff between recall and precision, we submitted additional runs that specifically aimed at higher precision or recall (see [9] for a description of these parameters). By default, the runs used gain >= 0.5, sentence length <= 20, top-5 and a relevance model over the past hour. Alternatively, the runs titled “docsRecall” and “IGnRecall” used length <= 30 and gain >= 0.3, and the run titled “IGn-Precision” used top-1 and a time window = 0.5 hour.

2.6 Efficiency

For the run titled “titles” the three exact nearest neighbors were retrieved for each news article title, however, for alternative approaches that consider the entire documents brute force retrieval of the nearest neighbors is no longer feasible. Therefore we approximated the three nearest neighbors for a given news article by indexing the trigrams and bigrams contained in each news article (ignoring stop words), and considering only articles that have at least one trigram in common, and if less than 10 such news articles are found also consider the articles that have at least one bigram in common.

3. RESULTS

In Table 1 are the results of our submitted runs and the average over all TREC participants for task 1 “Filtering and Summarization”. The original approach described in [9] was used in the run labelled “titles”, which has a lower comprehensiveness (recall) compared to the other runs. In fact, for 4 out of the 21 topics no results were returned simply because of the lack of news article titles that contain all query terms. Recall was improved in the run titles “docs” by matching the query terms in the entire document rather than just the title, however the gain (precision) is much lower possibly indicating that clustering the news articles using cosine similarity over their complete contents may add more useless documents than using only their titles. Following the findings in [7], we alternatively measured similarity between documents using normalized information gain in the runs that start with “IGn”, which obtains a consistently better gain-comprehensiveness trade-off. The run that specifically targets a higher gain “IGnPrecision” obtained the overall best performance in F-measure. Since the run titled “IGnRecall” was not annotated, the reported numbers are an estimation based on the overlap in results with the other runs and ignoring sentences that were not scored, which may not be accurate.

References

[1] 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.

[2] 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.

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

[4] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *SIGKDD*, 1999.

[5] 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.

[6] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[7] J. B. P. Vuurens, A. P. de Vries, R. Blanco, and P. Mika. Hierarchy construction for news summarizations. In *SIGIR TAIA Workshop*. SIGIR, 2015.

[8] J. B. P. Vuurens, A. P. de Vries, R. Blanco, and P. Mika. Online news tracking for ad-hoc information needs. In *ICTIR*. ACM, 2015.

[9] 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.

[10] 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.