Leveraging Learning To Rank in an Optimization Framework for Timeline Summarization

Giang Binh Tran, Tuan A. Tran, Nam-Khanh Tran,
Mohammad Alrifai and Nattiya Kanhabua
L3S Research Center, Hannover, Germany
{gtran, ttran,ntran,alrifai,kanhabua}@L3S.de

ABSTRACT

With the tremendous amount of news published on the Web every day, helping users explore news events on a given topic of interest is an acute problem. Timeline summaries have recently emerge as a simple and effective solution for users to navigate through temporally related news events. In this paper, we propose an optimization framework and demonstrate the use of Learning To Rank (LTR) to automatically construct timeline summaries from Web news articles. Experimental evaluations show that our approach outperforms existing solutions in producing high quality timeline summaries. We make our dataset publicly available for future research in the same area at `http://www.l3s.de/~gtran/timeline/`

1. INTRODUCTION

Due to the chronological characteristic of online news, timeline summarization has become a natural way to present a storyline. A timeline summary (TS or *timeline* in short) represents the development of a story over time by highlighting its most important events. As an example, during the active period of the Arab Spring revolution, one good timeline should capture events like "Egypt president Hosni Mubarak resigned after several protests on 11 February 2012.", "On 25 February, Libyan opposition claimed to control the power over Muammar Gaddafi.", etc. While news agencies often manually maintain timeline summaries for major events of wide community interest, constructing such summaries often requires considerable amount of human effort and does not scale well across different systems. An automated timeline summary generation is hence more advantageous at improving the user experience in online news exploration.

In this work, we address the issue of *automatically generating timeline summary* of Web news articles that are driven by a common topic. The challenges lie in extracting important points of the story that match common human need with minimal involvement of the user. While there has been a rich body of research in TS [15], [4], [2], to our knowledge none of existing approaches address using human timeline as supervision to improve the summarization quality. In this paper, we propose a novel framework to exploit human timelines to generate TS in a supervised manner. In particular:

TAIA'13 August 1, 2013, Dublin, Ireland.

(i) we propose criteria for measuring the quality of a TS and an optimization algorithm for producing high quality summaries.

(ii) we propose a framework that enables exploiting human timelines, thereby to avoid the cost for building training data. We apply a Learning To Rank method for ustilizing implicit information from human timelines to build a sentence ranking model and feed this to the optimization algorithm to create good timelines.

(iii) we provide experimental evaluation of the effectiveness of our approach against ones created by professional journalists, and in comparison with existing state of the art solutions.

2. RELATED WORK

Multi-document Summarization aims at extraction of information from multiple texts discussing the same topic. One popular extractive systems, the centroid-based multi-document summarizer MEAD [12] generates summaries by using information from a set of words that are statistically important to a cluster of documents for selecting sentences. News Blaster [9] clusters news into events and apply the MultiGen system to find sets of similar sentences; then cut and paste similar phrases to create a summary sentence for each set. There are several studies on supervised learning for query-based summarization, e.g. [11], where the authors trained regression models on a benchmark dataset to predict the relevance score for each sentence; then selected the top sentences to form up a query-based summary. However, such approaches do not cover the temporal aspects of a TS.

Timeline summarization There is a plethora of research in generating structured representation for news articles as a timeline summary. In most cases, a timeline generation system aims at extracting a set of *important* sentences and put them on a chronological time span [14], [2], [4], [15]. Swan and Allan [14] presented very first work on timeline overviews. They attempted to organize noun phrases and named entities extracted from specific time periods into stories. Allan et al. [2] relied on "usefulness", "novelty" of the sentences to provide summaries of news topics. They worked on stream of articles and aimed at including in a summary sentences with useful and novelty information. Yet, they did not deal with summarizing multiple events in documents to construct a timeline. Chieu et al. [4] used "interest" and "burstiness" scores, which intuitively indicate the popularity of the reported event in the sentence and the time point. Similarly, Yan et al. [15] employed "relevancy", "coverage", "coherence", "diversity" to create the timeline based on term frequency distribution. Our approach differs to previous work in two folds. First, we employ supervised *learning to rank* techniques to select sentences with minimum human effort on labelling training data. Second, we propose different criteria to characterize a good timeline summary.

3. TIMELINE SUMMARY MODEL

3.1 Key Concepts and Framework

News Events and News Topics: We focus on *news events* reported in web news articles, such as "Resignation of Egyptian president Mubarak". News events typically belong to the broader concept of a *news topic*, which spans longer interval of time, for example, the aforementioned news events can be associated to news topic "Egyptian revolution 2011". **Timeline Summaries**: A timeline summary is a compact chronological representation of a news topic. In this paper, we focus on summarizing news in daily basis. In this setting, a TS consists of a temporally ordered list of *day summaries*, where each day summary consists of a day and a (small) set of sentences describing the main news event(s) that occurred on that day.

Let $A_q = \{a_1, ..., a_{|A_q|}\}$ be the set of news articles related to a news topic q, and $A_q^{d_i} \in A_q$ be a subset of news articles that are published on day d_i. By S_{d_i} we denote the set of all sentences extracted from $A_q^{d_i}$. We let $TS_q = \{DS_{d_1}, ..., DS_{d_{|TS_q|}}\}$ be the timeline summary of q, where $DS_{d_i} \in TS_q$ is the day summary of the i-th day (d_i) in TS_q and $|TS_q|$ is the number of days in the timeline. Each day summary $DS_{d_i} = \{s_1, ..., s_{m_i}\}$ consists of a set of sentences describing the *main* events that occurred on day d_i. Note that DS_{d_i} is a subset of S_{d_i}, and $m_i \leqslant r$, where r is the maximum number of sentences in a day summary (in practice, this reflects a user-desired compression ratio).

TS Generation Framework: Our proposed framework generates a timeline summary in two steps. In *Sentence Ranking* step, we rank, for each day d_i, sentences in S_{d_i} by their content relevance to the day summary of d_i. In *Summary Optimization* step, for a given day, we select a maximum r sentences from its ranked sentence list to optimize the quality of the overall timeline summary.

3.2 Measuring the Quality of Day Summaries

The challenge of generating day summaries for a news topic q is in the selection of the subset of sentences DS_{d_i} for each day d_i in the timeline from the content of all articles $A_q^{d_i}$ published in that day. Therefore, we model the problem of generating day summaries as an optimization problem where the aim is to select the *best* subset of sentences DS_{d_i}. A formalization of the target function to be maximized three criteria content relevance, novelty and continuity.

Relevance: To produce a good summary for a day d, we need to predict the relevance of each sentence in the collection of news articles A_q^d that are published in day d. We model this task as a *ranking* task to deliver the top $k = r$ relevant sentences from the collection S_d of all sentences extracted from A_q^d. To this extent, we propose a learning to rank (**LTR**) approach to predict the relevance score $rel(s)$ of each sentence $s \in S_{d_i}$, which we describe in more details in Section 4.1. The overall relevance score of a day summary DS_{d_i} is computed as follows: $R(DS_{d_i}) = \sum_{j=1}^{m_i} rel(s_j)$.

Novelty: Novelty concept has been long studied in literature as a mean for increasing coverage and diversity of the returned items (e.g, [2, 15]). In our work, we define a sentence novelty by how much new information the sentence introduces compared to already selected sentences. Inspired by n-gram models which have been shown effective in measuring information overlapping [8], we formally define the novelty $nov(s, S)$ of a sentence s compared to a set of previously selected sentences S as the ratio of non-overlapping n-grams over total n-grams of s ($n = 2$ in our experiments): $nov(s, S) = \frac{|\{n-grams \notin (s \cap S)\}|}{|\{n-grams \in s\}|}$ The overall novelty score of a day summary DS_{d_i} is then computed as follows:

$N(DS_{d_i}) = \sum_{j=2}^{m_i} nov(s_j, \{\bigcup_{u=1}^{j-1}(s_u)\})$

Continuity: We have observed, from several expert-generated news topic timelines in newswire platforms, a *smooth transition* between events in consecutive days, i.e. descriptions of events in subsequent days tend to mingle with mentions of what happened previously that are related in a story line in order to make users recall better the overall picture. To simplify this characteristic in our model, we propose *continuity* as a measure for the connection degree between two subsequent day summaries. The measure of the continuity $con(s, DS_{d_{i-1}})$ for a sentence s from day d_i with respect to the previous day summary $DS_{d_{i-1}}$ is formally defined as follows: $con(s, DS_{d_{i-1}}) = \frac{|\{n-grams \in (s \cap DS_{d_{i-1}})\}|}{|\{n-grams \in s\}|}$. Finally, the overall continuity score of a day summary DS_{d_i} is defined as:

$C(DS_{d_i}) = \sum_{j=1}^{m_i} con(s_j, DS_{d_{i-1}})$

3.3 Problem Statement

We define the function (U) for measuring the quality of a day summary as a mingling function that balances the three attributes: sentence relevance (R), sentence novelty (N) and content continuity (C) as:

$U_{DS_{d_i}} = \lambda_1 * R(DS_{d_i}) + \lambda_2 * N(DS_{d_i}) + \lambda_3 * C(DS_{d_i})$

and the target function for maximizing the overall quality of the timeline summary as: $U_{TS_q} = \sum_{d_i \in TS} U_{DS_{d_i}}$

Given a collection of news articles A_q related to a news topic q and the required maximum number of sentences per day summary r, the goal of a timeline summarization system is to produce a timeline TS_q with the maximum value of U_{TS_q}.

4. TIMELINE SUMMARY GENERATION

4.1 Learning To Rank Sentences

Unlike unsupervised methods that compute the relevance of sentences by measuring their similarity to the news topic (e.g. [15]), we learn relevance function from human timeline summaries. Typically, LTR methods, like SVM-Rank [6] for example, learn a ranking function $h(F_s)$ that satisfies $h(F_{s_i}) > h(F_{s_j}) \iff rel(s_i) > rel(s_j)$, where $rel(s)$ is the relevance score of s and F_s is a vector of features extracted from s. For our training, we define $rel(s)$ is measured by comparing each sentence s in each cluster S_{d_i} with the sentences of the corresponding day summary DS_{d_i}. The intuition is that sentences from S_{d_i} that are similar to the ones in DS_{d_i} (which were selected by experts) are most relevant for the summary of d_i than the rest of S_{d_i}. However, in practice, human tends not to extract sentences from news to build a summary, but rather paraphrase and compact original sentences from news articles. Therefore, it is expected that sentences in DS_{d_i} does not exactly match ones in S_{d_i}. To cope with this problem, we propose measuring content similarity between any two sentences using method introduced in [10]. The measure exploits sentence-based TF.IDF weights and was proved to perform well in monolingual corpora[1] $sim(s_j, s_k) = \frac{\sum_t w_{s_j}(t).w_{s_k}(t)}{\sqrt{w_{s_j}^2(t) w_{s_k}^2(t)}}$. We then define the relevance score as: $rel(s) = \max_{k=1}^{|DS_i|} sim(s_j, s_k)$.

[1] t is a term occurring in both sentences s_j and s_k; $w_{s_j}(t)$ is sentence-based TF*IDF weight of term t , see [10] for the detailed description of how to compute $ws_j(t)$

4.1.1 Feature selection

For each sentence $s \in S_{d_i}$, we extract 28 features, which can be grouped into five different categories:

Surface-level features: we extract common surface-level features such as sentence length (in words), ratio of stop to non-stop words, number of pronouns, and $\frac{1}{pos}$ where *pos* is the position of the sentence in the article.

Coherence features: we consider some features that measure the coherence of the text: number of temporal relation signals (such as *when, since, before, etc.*), causal relation signals (such as *cause, lead to, etc.*) and logical relation signals (such as *in contrast, even though, etc.*). Studies in linguistics show that these signals create cohesive links between ideas and clauses [3] and thus suggest important information to be selected into the timeline summary.

Time-related features: We include binary feature *hasTempExp* to indicate whether a sentence has a temporal expression[2], since it has been shown effective in important event detection [7]. This category also includes the *popularity* proposed by [4]. Another feature is the number of words of the sentence s that are frequent words in articles published in the neighboring dates, for signifying the temporal relevance of the event(s) described in s in connection to the events of neighboring dates.

Topic features: includes some widely used such as the sum TF.IDF of words, top frequent words computed on the whole collection of related articles A_q; *cross-entropy* between the word distribution of the sentence s and that of the collection of news articles published at the day of focus d. Additionally, we measure the association of s and the date d by using log odds ratio [1], which performed well on summarization tasks as reported in [5].

In addition, we use a feature for measuring the similarity of the sentence s to the theme (i.e. the main content) of the article that includes s. We observe that news article typically starts with a short abstract describing the main event(s), followed by the main body, which provides more detailed information. Based on this observation, we select a small set of sentences $S_{abstract}$ from the beginning of the article (4 in our experiments) as a representative set of the main theme/content of the article. We then compute the similarity of a sentence s to this subset as follows:

$sim_{theme}(s) = \max_{s' \in S_{abstract}} sim(s, s')$

Event features: we use the feature mainEvent to measure the probability that a sentence s describes a main event of the day, as the bi-term similarity to list of most frequent pairs of words $TopPairs_d$ from the collection of articles of the day d. The intuition is that most frequent pair of words in this collection are likely to represent the main event(s) of the day. For example, on 11-Feb-2011, there is a main event that Egyptian president Mubarak resigned and handed over the power to the army. Therefore, it is likely that the pair of words "Mubarak" and "resign" often co-occur in many articles published on that day.

4.2 Summary Optimization

Given the set of sentences S_{d_i} from all articles of day d_i, we apply dynamic programming to solve the optimization problem for selecting the subset $DS_{d_i} \subset S_{d_i}$ that maximizes the target function $U_{DS_{d_i}}$ as described. The complexity of this Algorithm is $O(|S_{d_i}|^2)$ and it works as follows.

There are r slots for selecting r sentences to make the day summary DS_{d_i}. Let $U_{DS_{d_i}}[j][k]$ be the maximum target function if sentence $s_k \in S_{d_i}$ is selected for the j-th slot, and $H_i[j][k]$ be the current set of j selected sentences (hence, $s_k \in H_i[j][k]$).

[2]We use Heideltime toolkit[13] to parse the temporal expression in the sentences

The algorithm maximizes the following target function by looking at all possible selections for the $(j-1)$-th slot:

$U_{DS_{d_i}}[j][k] = \max\{U_{DS_{d_i}}[j-1][v] + \delta(s_k)\}$ such that s_k has not been selected before, i.e.: $s_k \notin H_i[j-1][v]$ and $\delta(s_k)$ denotes the added value to the target function when we choose the sentence s_k in the day summary and is computed by applying formula 3.3 on the sentence level for the local optimum:

$\delta(s_k) = \lambda_1 * rel(s_k) + \lambda_2 * nov(s_k, H_i[j-1][v]) + \lambda_3 * con(s_k, DS_{d_{i-1}})$

5. EVALUATION

5.1 Materials and Design

Since there has been no available dataset published for TS, we construct the evaluation dataset ourself and plan to publish our data for research purpose.

Collecting human timelines (ground truth): We collected available timelines published by popular news agencies such as CNN, BBC, NBCnews, etc. that discuss famous topics happening in recent years (e.g. "BP Oil Spill"). We only took English timelines where the timestamps are explicit dates, such as *07 July 2011* and ignored the timelines whose timestamps are at the year, month or week levels, such as "July 2006". As a result, we obtained 17 different timelines in **9** different topics: BP Oil, Michael Jackson Death, H1N1, Haiti Earthquake, Financial Crisis, Libyan War,Iraq War, Egyptian Protest.

Retrieving news articles: For each timeline, we used Google Web Search to retrieve news articles from the same agency of the timeline (i.e. BBC, CNN news articles for BBC, CNN-published timeline respectively) using topic news queries and time filter option and retained top 400 returned articles that are published during the timeline timespan. At the end, we obtained 4650 news articles after duplication removal. Then, we used BoilerPipe[3] and additional hand-crafted rules to extract the content of the news articles. Data set are published online at: `http://www.l3s.de/~gtran/timeline/`.

Training and Testing data: To evaluate our framework, we conduct experiments with 9-fold cross-validation based on topics. At each round, all models have been trained using timelines of 8 topics, and separately tested on timelines of 1 topic left [4]

5.2 Experiment

Comparison setting We evaluate our system against traditional multi-document summarization and timeline generation systems. The day summary length of all systems' output is forced to the parameter r , which is computed as the rounded value of average sentences per date from the ground truth data.

Random The system generates day summary for a date d_i by randomly selecting sentences for S_{d_i}.

MEAD[5] is a multi-document summarization system proposed by Radev et al. [12] implemented centroid-based approach and is then enhanced with various of features later.

Chieu et al.[4] is another multi-document summarizer which utilizes the popularity of a sentence as TFIDF similarity with other sentences to estimate its importance. We use the same settings reported in [4].

ETS is by far the best TS system in news domain. We implemented the ETS algorithm described in [15] with the same setting. For fair comparison, we sampled our data and contact the authors to verify the discrepancy in the two outputs.

[3]http://www.l3s.de/ kohlschuetter/boilerplate/

[4]Different to the experiment setting in [15], we don't mix the timelines of a same topic together. Our assumption is, journalist created timeline from articles of their news agency.

[5]We used MEAD 3.12, http://www.summarization.com/mead/

Our proposed approach We use SVM-rank to demonstrate the performance of our system, which is one of the most common LTR implementations and has been widely used in many IR tasks.

Evaluation Metric: We noticed that human generated timelines are inherently varying, and often subjective, making it difficult to rely on user evaluations. Hence, we used the metric based on the ROUGE scores that are widely used in traditional summaization tasks. In timeline evaluation tasks (e.g. [15]), the quality of different TSs are compared via F-measure of the ROUGE-1, ROUGE-2. In this paper, we adopted the same metrics, plus the additional ROUGE-S*. Technically, ROGUE-S* is computed the same as bigram-based ROGUE-2 scores, but it allows the words in the bigram to be aparted by a window. This makes ROGUE-S* capture better the global distributional semantics, while traditional ROGUE-Ns capture better the local semantics, i.e. sentence to sentence matching.

Result: The average results of TS generation on our dataset are represented in Table 1. Overall, our system using SVM-rank obtains the best results, followed by the ETS then Chieu et al., and it is no surprise that Random provides the worst results. Our system is better than Chieu et al.'s method at all scores. This is due to the fact that Chieu et al.'s method relies on sentence similarity over date-based TF.IDF, and thus fails to capture the semantic relations between events in the sentences. This becomes clear when Chieu et al.'s method gains better ROUGE-S* scores but worse ROUGE-1 and ROUGE-2 than MEAD. Similarly, ETS is better than Chieu et al. since it captures semantic aspects by reducing the gap between word distributions of the timeline summary and of the news.

Table 1: Average results on 17 timelines, the reported results are computed 95% confidence interval

	ROUGE-1	ROUGE-2	ROUGE-S*
Random	0.128	0.021	0.026
Chieu et al.	0.202	0.037	0.041
MEAD	*0.208*	*0.049*	0.039
ETS	0.207	0.047	*0.042*
Ours System	**0.230**	**0.053**	**0.050**

Our system performs better than MEAD in the experiments, even when we use advanced MEAD settings [6]. The reason is that MEAD treats documents of different days separately, thus it fails to capture global semantics as well as temporal information.

ETS system is the closest one to ours in that we also target local and global semantics (through our 3 criteria), but our system produces timeline summaries higher in all scores. The reason is we leverage some latent factors under supervision of human timelines, which is likely to be more advantageous than the unsupervised manner. In addition, we exploit event-oriented aspects and coherence relations through corresponding features, and through the optimization afterwards. Intuitively, these features contribute well in the timeline summarization task, since they reflect better the event semantics rather than word distribution distances as used in ETS.

Effects of Criteria In addition to the above comparison with other methods, we investigate how each criterion benefits timeline modeling. Since relevance score is essential in our model, we only study on the impact of the other criteria on the performance. We first study the benefit of novelty by varying the corresponding parameters λ_1 from 0.1 to 1.0 with a step-size of 0.05, keeping $\lambda_2 = 1 - \lambda_1$ and measuring the performance of the system on all 17 timelines with the leave-one-out strategy. The average best result (ROGUE-1 = 0.227; ROGUE-2 =0.052; ROGUE-S* = 0.048) is obtained when $\lambda_1 = 0.6, \lambda_2 = 0.4$, indicating that proposed novelty criterion can benefit the timeline summarization task. Next, we examine benefit of continuity to the day summary. Similarly, we vary λ_3 from 0.0 to 1.0 with a step-size of 0.05 while keeping $\lambda_1 = 1 - \lambda_3$. We obtain the best result at $\lambda_1 = 0.5$, $\lambda_2 = 0.5$ with (ROGUE-1 = 0.226; ROGUE-2 =0.053; ROGUE-S* = 0.049), indicating that proposed continuity criterion can benefit the timeline summarization task. For final result, we use the linear ratios $\lambda_1 : \lambda_2 = 3 : 2$ and $\lambda_1 : \lambda_3 = 1 : 1$ as achieved above.

[6]We use advanced features in producing query-based summary on the top of surfaced and centroid-based model, see MEAD documentation for more details

6. CONCLUSION

In this paper, we present a novel approach to automatically construct timeline summary from a collection of news articles orienting to a given news topic. Our framework utilizes machine learning approach and dynamic programming to create a good timeline summarization; propose un-biased criteria that can model timeline's characteristics, namely relevance, novelty and continuity. We developed a corpus consists of 17 expert generated timelines and correlated news articles that belongs to 9 topics and do evaluation this corpus. Our experimental results showed that our method performs better than current states-of-the-art. For future work, we plan to design features that can avoid inclusions of future data into the summarization and improve presented criteria computation.

7. REFERENCES

[1] A. Agresti. Introduction to categorical data analysis. *John Wiley and Sons, New York.*, 1996.

[2] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *Proceedings of SIGIR'01*, pages 10–18, 2001.

[3] K. Cain and H. M. Nash. The influence of connectives on young readers' processing and comprehension of text. *Journal of Educational Psychology*, 103:429–441, 2011.

[4] H. L. Chieu and Y. K. Lee. Query based event extraction along a timeline. In *Proceedings of SIGIR'04*, pages 425–432, 2004.

[5] S. Fisher and B. Roark. Query-focused supervised sentence ranking for update summaries. In *TAC-2008*, 2008.

[6] T. Joachims. Training linear svms in linear time. In *KDD*, pages 217–226, New York, NY, USA, 2006. ACM.

[7] N. Kanhabua, S. Romano, and A. Stewart. Identifying relevant temporal expressions for real-world events. In *Proceedings of TAIA workshop, SIGIR*, 2012.

[8] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NAACL'03 - Volume 1*, pages 71–78, 2003.

[9] K. McKeown, R. Barzilay, J. Chen, D. K. Elson, D. K. Evans, J. Klavans, A. Nenkova, B. Schiffman, and S. Sigelman. Columbia's newsblaster: New features and future directions. In *HLT-NAACL*, 2003.

[10] R. Nelken and S. M. Shieber. Towards robust context-sensitive sentence alignment for monolingual corpora. In *In EACL*, 2006.

[11] Y. Ouyang, S. Li, and W. Li. Developing learning strategies for topic-based summarization. In *CIKM*, pages 79–86, New York, NY, USA, 2007. ACM.

[12] D. R. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. ÃĞelebi, S. Dimitrov, E. DrÃąbek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang. Mead - a platform for multidocument multilingual text summarization. In *Proceedings of LREC'04*, 2004.

[13] J. Strötgen and M. Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of SemEval'10*, pages 321–324, 2010.

[14] R. Swan and J. Allan. Automatic generation of overview timelines. In *Proceedings of SIGIR'00*, pages 49–56, 2000.

[15] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of SIGIR'11*, pages 745–754, 2011.