A Dynamic Evolutionary Framework for Timeline Generation based on Distributed Representations

Dongyun Liang, Guohua Wang, Jing Nie
Tencent
dylanliang@tencent.com

ABSTRACT

Given the collection of timestamped web documents related to the evolving topic, timeline summarization (TS) highlights its most important events in the form of relevant summaries to represent the development of a topic over time. Most of the previous work focuses on fully-observable ranking models and depends on hand-designed features or complex mechanisms that may not generalize well. We present a novel dynamic framework for evolutionary timeline generation leveraging distributed representations, which dynamically finds the most likely sequence of evolutionary summaries in the timeline, called the Viterbi timeline, and reduces the impact of events that irrelevant or repeated to the topic. The assumptions of the coherence and the global view run through our model. We explore adjacent relevance to constrain timeline coherence and make sure the events evolve on the same topic with a global view. Experimental results demonstrate that our framework is feasible to extract summaries for timeline generation, outperforms various competitive baselines, and achieves the state-of-the-art performance as an unsupervised approach.

CCS CONCEPTS

• **Information systems** → *Information retrieval diversity*; *Retrieval tasks and goals*; Summarization.

ACM Reference Format:
Dongyun Liang, Guohua Wang, Jing Nie. 2019. A Dynamic Evolutionary Framework for Timeline Generation based on Distributed Representations. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Along with the rapid growth of the World Wide Web, there is a huge document collection related to the news topics. General search engines rank the indexed document according to their understanding of the user's query relevance. and users have access to a complete range of ranked documents about a particular topic. However, users still possibly get lost in redundant results, even if the ranked documents have been ordered by time. It is significant to provide a timeline for users to view what evolutionary topic is going on and what key events break out along with a particular topic.

As handcrafted timeline requires tremendous human labor of reading and understanding, timeline summarization (TS) is a widely adopted task to generate timelines [7, 8, 17]. News topic can be broken into a sequence of events, and TS provides temporal summaries of the evolution of news events related to the topic. Given news marked date, such as the indexed date by search engine in practice, we aim to tackle this problem for selecting a subset of important dates as the major points along the timeline [5], meanwhile, generating a good daily summary for these dates [13]. Some researches [1, 10] use clustering and ranking techniques to select the important events that be included in the final summary, and most of the previous work relies on hand-designed features [3, 18] or complex mechanisms.

In recent years, several distributed representations approaches [2, 11] caused the trend of deep learning of the text. More and more embeddings derived from deep network for word or sentence have been proved efficient in representing rich syntactic and semantic information, which can somewhat help people free from the tedious feature engineering. There have been some efforts that explores this in TS [16, 19]. Distributed representation of a target word or sentence derives from the association of adjacency [6], which carries around information to achieve semantic coherence and is conceptually similar to the coherence of adjacent event and the relevance on a same topic in TS. However, as far as we know, most of the explorations in TS treat representations as a extra feature to enhance the effect [16], there is no framework which is a natural use of distributed representations to integrate their adjacency relevance into the coherence of TS.

In this paper, we propose a novel framework to generate timeline, which is dynamic and evolutionary to make natural use of distributed representations. In specific, we assume that the timeline has a certain coherence [18]. That is, with the evolution of the topic, there is not necessarily a direct relevance between the first event and the last rather than the next event, but the adjacent events in the timeline have a certain relevance in the news reports. Since reporting a new event on a same topic often cites the recent event reports, coherent events always have partial similarities in text, and this effect will gradually weaken as the timeline develops. We also assume that the timeline has a global view to guide the central theme [17]. As the events are assigned to a particular topic, the events along the timeline are more or less related to this topic.

Our approach takes a dated collection relevant to a news topic as input, and output a timeline with the component summaries of events which represent evolutionary trajectories on specific dates. Firstly, we perform embedding learning to model the continuous vector representations of the inputs. Then we cluster massive amounts of news paragraphs into event groups according to their representations and the corresponding date, and we get several

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

event groups in each step by date. Under the guidance of coherence and global view, the framework try to find a dynamic optimal path linking the events between these steps. The experimental results show significant and consistent performance improvement over the state-of-the-art methods on public datasets.

2 RELATED WORK

There have been many studies about timeline generation from various sources, including the sentences, paragraphs or headlines of news. [3] extracted the popular and bursting sentences to place along the timeline from a query. [18] optimized the problem via iterating substitution by incorporating several constraints. Supervised learning [14] is also widely used in TS, and [15, 17] proposed a ranking framework to get temporal summarization. Most of summarization corpora are text-only, and [16] utilized both text and image to provide a comprehensive sketch of the topic evolution. Considering timeline as latent variables, many dynamic approaches [8, 9] based on probabilistic graphical models have been proposed to discover the evolving patterns. [13, 14] published the datasets that consists of the timelines created by experts, the correlated news articles and headlines.

Some work [1, 12] has focused on a better understanding of a particular entity or event by displaying a list of episodes in time order, and they jointly consider the relevance and temporal diversity to interpret the cause and effect of the entity. Another related work is concerned on text stream summarization: [10] discovered key information in vast text, such as the events from trending and breaking news, then organized that. There exists explorations [7] for individual timeline from Twitter. However, hand-designed features account for a large proportion in the above work, which may not generalize well. As deep learning has gained immense success on Natural Language Processing, [16, 19] introduce distributed representations into TS.

3 FRAMEWORK

3.1 Preliminaries

In our work, we focus on the news topic, such as *2010 British Oil spill*, and the events that evolve with the development of topics. TS consists of a temporally ordered list of summaries, which describes the main events that occurred along the time.

Let $D = \{D_1, D_2, \ldots, D_T\}$ is the set of news documents related to a particular topic q, where $D_i \in D$ is the subset of news collected on the period of i-th day. We extract the paragraphs $A_i = \{A_i^1, \ldots, A_i^m\}$ from D_i to denote the candidates of the TS , which can be the news headline, first sentence and n-gram sentences of the content. $TS_q = \{A_{t_1}^{m_1}, A_{t_2}^{m_2}, \ldots, A_{t_{|S|}}^{m_{|S|}}\}$ denotes the TS about q, where $A_{t_i}^{m_i} \in A_{t_i}$ is the summary of the t_i-th day in TS, $t_i \in \{1, \ldots, T\}$ is the specific date on the period of TS, m_i denotes the specific sample of A_{t_i} in t_i-th day , and $|S|$ is the total steps of the timeline.

3.2 Learning Distributed Representations

There are various methods to get distributed representations for short text. To learn the representations for the paragraphs, we filter stop words out of them, use skip-gram model [2] to learn vector embeddings of the words, and multiply them with their TF-IDF scores to represent the paragraphs, which has been proven to be a baseline [4] and feature across a multitude of tasks, especially short text similarity tasks.

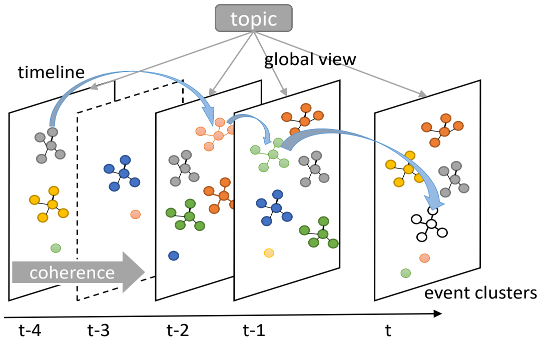

Figure 1: Dynamic Evolutionary timeline sequence of event clusters guided by coherence and global view.

Given the topic q and the dated news documents D related to q that can be obtained by the retrieval recall, we can optionally expand q by some keywords or descriptive text, such as Wikipedia knowledge returned by search engine, and extract paragraphs subset A_i from D_i. As mentioned above, q is embedded to a vector $v(q)$, and A_i^j is embedded to a vector $v(A_i^j)$.

In reality, there are many repeated reports about the same event in redundant web text. Hence, taking the paragraphs from A as a whole, we separate the candidates A_i^j into disjoint clusters in embedding space by affinity propagation algorithm, obtain the number K of clusters $C = \{C_1, \ldots, C_K\}$, and then divide the clustering result into day by clipping the majority vote of the date. The event clusters of i-th day and j-th class is denoted as:

$$event_{i,j} = \bigcup^{j \in C_j} A_i^j, \quad 1 \leq i \leq T \; and \; 1 \leq j \leq K \tag{1}$$

The timeline is the most likely sequence of event clusters, called the Viterbi timeline, shown in Figure 1. We need to find out the dynamic path linking the event clusters as sequence TS_{event}:

$$TS_{event} = \bigcup_{(i,j) \in X} event_{i,j} \tag{2}$$

$$X = [(t_1, n_1), (t_2, n_2), \ldots, (t_{|S|}, n_{|S|})]$$

where n_i is the specific class of event clusters in t_i day, and TS_q will be further extracted from TS_{event}. each embedding of event cluster is defined as the mean value of the candidates A_i^j in the same cluster at vector dimension:

$$v(event_{i,j}) = \begin{cases} \overrightarrow{0}, & \text{if } \bigcup^{j \in C_j} A_i^j = \emptyset \\ \text{AvgPooling} \bigcup^{j \in C_j} v(A_i^j), & \text{otherwise} \end{cases} \tag{3}$$

3.3 Viterbi Timeline

The Viterbi timeline derives from a origin path $X' = [(1, n_1), \ldots, (T, n_T)]$, which is a sequence of clusters (i, n_i) that generates the event timeline, where $n_i \in \{C_1, \ldots, C_K\}$. Two 2-dimensional tables of size $T \times K$ (W_1 and W_2) are constructed as follows.

Each element $W_1[i,j]$ of W_1 stores the weight of the most likely path so far $[n_1,\ldots,n_i]$ with $n_i = C_j$ that generates event clusters path. Each element $W_2[i,j]$ of W_2 stores $i-1$ of the most likely path so far $[(1,n_1),\ldots,(i,n_i = Cj)]$. The table entries $W_1[i,j]$ and $W_2[i,j]$ are filled by increasing order of $K \cdot i + j$.

$$W_1[i,j] = \max_k (W_1[i-1,k] \cdot Q^i_{kj} \cdot R_{ij}) \tag{4}$$

$$W_2[i,j] = \text{argmax}_k\, (W_1[i-1,k] \cdot Q^i_{kj} \cdot R_{ij}) \tag{5}$$

with the dynamic changes of k, R_{ij} plays the role of global view, and Q^i_{kj} leverages the adjacent relevance to preserve the coherence, such as defined below.

Coherence. The size of Transition matrix Q is $T \times K \times K$, such that Q^t_{ij} stores the transition weight of transiting from cluster C_i to cluster C_j at the time t:

$$Q^t_{ij} = \text{Cosine}(v(event_{t-1,i}), v(event_{t,j})) \tag{6}$$

transition weight is used to measure the relevance of the event clusters between previous and present time status in the consecutive timeline. Timeline is optimally solved by breaking it into sub-relevance and then recursively finding the optimal coherence to the global relevance, which meets the assumption of coherence.

Global View. The size of Emission matrix R is $T \times K$, such that R_{ij} stores the weight of topic q expression from cluster C_j at time i:

$$R_{ij} = \text{Cosine}(v(q), v(event_{i,j})) \tag{7}$$

Given a underlying topic, emission weight represents how likely each possible event cluster is along with timeline, which impacts the expression of events in global view and generates the thread running through many of these events.

Temporal association of timeline are the temporal constraining local correlation, and the timeline takes a global view of the related events. Local correlation Q and global view R are interrelated with each other. The entire procedure is summarized in Algorithm 1.

3.4 Constraints

Since the timeline is not necessarily continuous by days, some time-windows may be filled with the event clusters irrelevant to the timeline. In addition, the event will be reported again by many medias after its first burst, bringing some repeated reports. We propose the operation (A) in Algorithm 1 to reduce the impact of irrelevance and repeat.

Relevance α is a measure of continuity at current day, denotes elementwise multiplication of two matrices, and coefficient of variation[1] c_v is a standardized measure of the burst of the news at that day. If the adjacent relevance α is too weak, it means that all events at current step can not properly undertake the above information. With the benchmark of $v(q)$, each news is treated as a sample value at the distribution, and the c_v gives a degree of dispersion about the collected news at that step. As showed in operation (A), we rely on experience to set the hyper-parameters β_α and β_{c_v}. Once the conditions are met, the previous state of W_1 and W_2 will be retained, The W_2 path will be filled with -1 to represent jumping over this step. To address the final TS, we filter $(i, n_i = -1) \in X'$ to get the Viterbi timeline X, and extract each $A^{m_i}_{t_i}$ to generate TS_q:

$$m_i = \text{argmax}_j\, \text{Cosine}(v(q), v(A^j_{t_i})), \quad A^j_{t_i} \in event_{t_i,n_i} \tag{8}$$

ALGORITHM 1: Capturing the origin path X'

```
Function generate Viterbi timeline
    foreach cluster $j \in \{1,\ldots,K\}$ do
        $W_1[1,j] \leftarrow \text{Cosine}(v(q), v(event_{1,j}))$;
        $W_2[1,j] \leftarrow 0$;
    foreach date $i \in \{1,\ldots,T\}$ do
        $\alpha = \max_j \max_k Q^i_{kj} \circ R_{kj}$;
        $c_v = \frac{\sigma(v(A_i)^{\mathrm{T}} v(q))}{\mu(v(A_i)^{\mathrm{T}} v(q))}$ ;
(A)     if $\alpha < \beta_\alpha$ or $c_v > \beta_{c_v}$ then
            foreach cluster $j \in \{1,\ldots,K\}$ do
                $W_1[i,j] \leftarrow W_1[i-1,j]$;
                $W_2[i,j] \leftarrow -1$;
                update $Q^{i+1}_{:,j}$ for $event_{i,j} = \bigcup^{j \in C_j} A^j_{i-1}$;
            continue;
        foreach cluster $j \in \{1,\ldots,K\}$ do
            $W_1[i,j] \leftarrow \max_k (W_1[i-1,k] \cdot Q^i_{kj} \cdot R_{ij})$;
            $W_2[i,j] \leftarrow \text{argmax}_k (W_1[i-1,k] \cdot Q^i_{kj} \cdot R_{ij})$;
    $z_T \leftarrow \text{argmax}_k (W_1[T,k])$;
    $n_T \leftarrow C_{z_T}$;
    for $i \leftarrow T, T-1, \ldots, 2$ do
        $z_{i-1} \leftarrow W_2[i, z_i]$;
        $n_{i-1} \leftarrow C_{z_{i-1}}$;
    return $X'$
```

4 EXPERIMENTS

We experiment with two public datasets that have been proposed to investigate the timeline.

17 Timelines [14]. The dataset includes 17 timelines published by the major news agencies, such as CNN, BBC, and NBC News. They developed from 9 different topics, including BP Oil, Michael Jackson Death, H1N1, Haiti Earthquake, Financial Crisis, Libyan War, Iraq War and Egyptian Protest. Each timeline has its own independent documents set related to the corresponding topic, and It overall contains 4,650 news documents of which the timestamps are explicit dates, such as 07 July 2011.

Crisis data [13]. It includes four crisis topics (wrt. Egypt, Libya, Yemen, Syria), and each topic has around 4,000+ documents with date timestamps. The timeline under the same topic has the same document set, which consist of the content and headline of the news articles. There are totally 25 manually created timelines for these topics. The headlines are the best summarization for the news, so we focus on headlines timeline on this dataset, rather than extract sentences from the news documents as candidates.

We compare our proposed framework with these baselines:

- **Random**: sentences are randomly selected as TS.
- **Chieu et al.** [3]:a multi-document summarizer which utilizes the popularity of a sentence as TF-IDF similarity with other sentences to estimate its importance.

[1] https://en.wikipedia.org/wiki/Coefficient_of_variation.

Table 1: Performance of models on 17 Timelines

Methods	ROUGE-1	ROUGE-2	ROUGE-S
Random	0.128	0.021	0.026
Chieu et al.	0.202	0.037	0.041
ETS	0.207	0.047	0.042
Tran et al.	0.230	0.053	0.050
Regression	0.303	0.078	0.081
Wang et al.	0.312	0.089	**0.112**
Ours	**0.334**	**0.105**	0.103

Table 2: Performance of models on Crisis data

Methods	ROUGE-1	ROUGE-2	ROUGE-S
Regression	0.207	0.045	0.039
Wang et al.	0.211	0.046	0.040
Ours	**0.268**	**0.057**	**0.054**

- **ETS** [17]: a unsupervised TS system in news domain.
- **Tran et al.** [14]: a system based on learning to rank techniques, the earliest baseline reported on the 17 Timelines dataset.
- **Regression** [15]: a supervised regression model to extract sentence as summarization.
- **Wang et al.** [16]: a low-rank approximation based approach that leverage the matrix factorization techniques and treat the multi-document extractive summarization task as a sentence recommendation problem.

We use common summarization metrics (F-measure of ROUGE) [18] to evaluate the quality of the TS generated by models. The system summarie would be individually evaluated against all reference summaries for the same topic on Crisis data, but against them for the same topic and news agencies on 17 Timelines. As the standard evalution of the datasets instructs, we treat each sentence as a candidate on 17 Timelines, and adopt each headline of the news on Crisis data. To learn the distributed representations, we use pre-trained word vectors[2], trained on Common Crawl and Wikipedia by fastText tookit [2]. Furthermore, each $v(q)$ is only embedded by the name of the topic q as a experimental control.

Table 1 show the performance of all models on the 17 Timelines dataset. We can see that our proposed approach is quite comparable to other state-of-the-art models. It beats others by a large margin by ROUGE-1 and ROUGE-2. Chieuet al and ETS gives the lower F score, indicating too much handcraft features to constrain TS is restrictive. Tran et al and Regression are a typical idea of learning to rank, and Wang cast it as a sentence recommendation problem by matrix factorization. Thought they utilize the supervised information to help the model learning timeline rules, however, supervised model for timeline generation has its insufficiency that the development of timeline about different topic are many and varied. Our assumption about timeline is well reflected to achieve an ingenious combination of dynamic evolution and distributed representations in the entire framework. The best result we obtain demonstrates that the novel union of distributed representations can benefit the TS task.

We report the results of our framework and the baselines on Crisis data in Table 2. Regression method is shown as a strong supervised baseline, and Wang's method is the past state-of-the-art method on this dataset. They don't explicitly consider much natural feature of the timeline, such as coherence and overall, thus fails to capture the semantic relations between events in short titles. We lead throughout the timeline by the coherence of the events and a global view of the topic. The same hyper-parameters as 17 Timelines are set, and it shows that Viterbi timeline can improve the performance significantly in general.

5 CONCLUSIONS

In this work, we propose a dynamic evolutionary framework for timeline generation, which addresses concerns over events on both coherence and overall. At its heart, we propose the Viterbi timeline, and it actually generate the natural association of TS with distributed representations. Experiments on 17 Timelines and Crisis data demonstrate the effectiveness of the TS framework. In the future, we would like to base the past and future contexts to generate a finite event sequence as timeline.

REFERENCES

[1] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. 2015. TimeMachine: Timeline generation for knowledge-base entities. In *In SIGKDD*. ACM, 19–28.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *In TACL* (2017), 135–146.

[3] Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *In SIGIR*. ACM, 425–432.

[4] Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. *In ACL* (2016).

[5] Remy Kessler, Xavier Tannier, Caroline Hagege, Véronique Moriceau, and André Bittar. 2012. Finding salient dates for building thematic timelines. In *ACL*.

[6] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*. 2177–2185.

[7] Jiwei Li and Claire Cardie. 2014. Timeline generation: Tracking individuals on twitter. In *In WWW*. ACM, 643–652.

[8] Jiwei Li and Sujian Li. 2013. Evolutionary hierarchical dirichlet process for timeline summarization. In *In ACL*. 556–560.

[9] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2016. Dynamic clustering of streaming short documents. In *In SIGKDD*. 995–1004.

[10] Bang Liu, Di Niu, Kunfeng Lai, Linglong Kong, and Yu Xu. 2017. Growing Story Forest Online from Massive Breaking News. In *In CIKM*. ACM, 777–785.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).

[12] Arunav Mishra and Klaus Berberich. 2016. Event digest: A holistic view on past events. In *In SIGIR*. ACM, 493–502.

[13] Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *In ECIR*. Springer, 245–256.

[14] Giang Binh Tran, Tuan A Tran, Nam-Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. Leveraging learning to rank in an optimization framework for timeline summarization. In *SIGIR TAIA*.

[15] Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. *In NAACL* (2015).

[16] William Yang Wang, Yashar Mehdad, Dragomir R Radev, and Amanda Stent. 2016. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In *In NAACL*. 58–68.

[17] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *In EMNLP*. 433–443.

[18] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *In SIGIR*. ACM, 745–754.

[19] Deyu Zhou, Linsen Guo, and Yulan He. 2018. Neural Storyline Extraction Model for Storyline Generation from News Articles. In *In NAACL*. 1727–1736.

[2] https://github.com/facebookresearch/fastText