Summarize Dates First: A Paradigm Shift in Timeline Summarization

Summarize Dates First: A Paradigm Shift in Timeline Summarization

Moreno La Quatra
moreno.laquatra@polito.it
Politecnico di Torino
Turin, Italy

Luca Cagliero
luca.cagliero@polito.it
Politecnico di Torino
Turin, Italy

Elena Baralis
elena.baralis@polito.it
Politecnico di Torino
Turin, Italy

Alberto Messina
alberto.messina@rai.it
Radiotelevisione Italiana (RAI),
Centre for Research and
Technological Innovation
Turin, Italy

Maurizio Montagnuolo
maurizio.montagnuolo@rai.it
Radiotelevisione Italiana (RAI),
Centre for Research and
Technological Innovation
Turin, Italy

ABSTRACT

Timeline summarization aims at presenting long news stories in a compact manner. State-of-the-art approaches first select the most relevant dates from the original event timeline then produce per-date news summaries. Date selection is driven by either per-date news content or date-level references. When coping with complex event data, characterized by inherent news flow redundancy, this pipeline may encounter relevant issues in both date selection and summarization due to a limited use of news content in date selection and no use of high-level temporal references (e.g., the past month). This paper proposes a paradigm shift in timeline summarization aimed at overcoming the above issues. It presents a new approach, namely *Summarize Date First*, which focuses on first generating date-level summaries then selecting the most relevant dates on top of summarized knowledge. In the latter stage, it performs date aggregations to consider high-level temporal references as well. The proposed pipeline also supports frequent incremental timeline updates more efficiently than previous approaches. We tested our unsupervised approach both on existing benchmark datasets and on a newly proposed benchmark dataset describing the COVID-19 news timeline. The achieved results were superior to state-of-the-art unsupervised methods and competitive against supervised ones.

CCS CONCEPTS

• **Information systems** → **Summarization**; • **Computing methodologies** → **Machine learning algorithms**.

KEYWORDS

Timeline summarization; Natural Language Processing; COVID-19 timeline

SIGIR '21, July 11–15, 2021, Virtual Event, Canada.

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462954

ACM Reference Format:
Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. Summarize Dates First: A Paradigm Shift in Timeline Summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462954

1 INTRODUCTION

Long-lasting news topics such as the outbreak of the Sars-Cov-2 pandemic and the 2020 U.S. elections have captured the attention of the main newspapers and online broadcasters for a long period. This has produced a huge volume of on-topic news articles that can be retrieved and analyzed. The ubiquity of complex event data over the Web makes the problem of extracting succinct descriptions of long-lasting news topics particularly challenging. News timelines must be highly relevant, minimally redundant, and with minimal latency, thus embedding timeliness information [28].

TimeLine Summarization (TLS) aims at extracting timestamped summaries on long-lasting news topics [31]. To help newspaper readers to keep track of long-lasting news stories, TLS approaches focus on extracting a temporarily ordered selection of news articles' content. Unlike temporal summarization [1], TLS assumes that the input news data stream is already filtered by focusing on portions of raw data that are relevant to the main event topic.

TLS is commonly addressed as a two-step process. The first one entails identifying the key dates in the news timeline, namely the *date selection* step. The second extracts the most relevant content (typically, a selection of news sentences) from the news articles relative to the previously selected dates, i.e., the *date summarization* step. These steps can be addressed either as separate tasks, i.e., date selection followed by date summarization on top of the selected dates (e.g., [10]), or as a unified process (e.g., [18]).

Challenges. The main issues encountered by state-of-the-art approaches are enumerated below.

- *Limited use of news content in date selection.* State-of-the-art date selection approaches do not (or minimally) rely on *content-based strategies*, instead focusing on analyzing date-level references. Furthermore, the inherent redundancy of

the input news flows makes current content-based similarity models prone to errors. Therefore, there is a need for new date selection strategies able to effectively exploit both *content* and *temporal references*.
- *No use of high-level temporal references.* Best performing date selection strategies rely on the analysis of date-level references (e.g., a news article that cites a past event occurred on a specific date). However, to the best of our knowledge, they currently disregard high-level references (e.g., a news article citing the events occurred in a past month). Notably, the number of occurrences of high-level references in the benchmark datasets (see Table 2) is higher than those of date-level ones. This leaves room for substantial model enrichment.
- *Need for periodic timeline updates.* News timelines need to be frequently updated until the long-lasting event ends. However, in the existing TLS pipeline each update requires to recompute the news summaries of all the relevant dates. Thus, frequent updates can be computationally expensive. This calls for more efficient, incremental timeline summarization strategies.

To tackle the above issues, this paper proposes a paradigm shift in TLS. It consists in reversing the two-step TLS process from the traditional pipeline (i.e., *select dates first and then summarize*) to *Summarize Dates First* (SDF, in short) and then select the key dates on top of summarized knowledge.

Contribution. The main contributions of the present work are detailed below.

- *New TLS pipeline.* We *first* generate summarized versions of the per-date news content, which can be conveniently exploited to drive the date selection process. To this purpose, SDF leverages summarized knowledge in content-based date selection. As shown in Section 5, exploiting a less redundant, summarized version of the per-date content significantly improves TLS performance.
- *Exploitation of high-level temporal references.* The date selection step in SDF effectively combines content-based analyses with a multiple-level date reference model. More specifically, the proposed approach does not only rely on date-level references, but also on the high-level ones neglected by previous approaches. For example, when a news article published in November 2020 includes an explicit reference to December 2019 (i.e., the beginning of the Sars-Cov-2 pandemic in China), such information is deemed as relevant to reward the dates within the referenced time period.
- *Incremental approach.* SDF efficiently supports frequent timeline updates. When the updating frequency is relatively high, SDF requires to recompute a more limited number of new summaries. Conversely, the traditional pipeline requires to extract again the summaries of all the relevant dates. Since the summarization process is the most computationally expensive step (see Section 3.4), this yields a relevant efficiency improvement.
- *New benchmark dataset.* State-of-the-art TLS algorithms were tested on three main benchmark event datasets (i.e., [10, 30, 32]). However, they show rather variable performance depending on the dataset characteristics. To strengthen the research findings in TLS, we release a new benchmark tailored to the TLS task, which describes the Sars-Cov-2 pandemic timeline. For validation purposes, the dataset is associated with a humanly generated public ground truth[1].

To compare the performance of the proposed approach with state-of-the-art methods, we conducted an extensive empirical evaluation. The results show that SDF, which relies on a unsupervised summarization pipeline, performs significantly better than all unsupervised methods and is competitive against supervised approaches (which require the availability of on-topic training data).

The remainder of the paper is organized as follows. Section 2 discusses the position of the current work in the state of the art. Section 3 thoroughly describes the SDF approach. Sections 4 and 5 respectively describe the benchmark datasets and summarize the main experimental results. Finally, Section 6 draws conclusions and discusses the future research agenda.

2 RELATED WORKS

A relevant research effort has been devoted to Timeline Summarization (TLS). Table 1 reports a categorization of the previous studies, sorted by publication date. Beyond paper citation and publication year, for each study we report (i) the process type (supervised or unsupervised), (ii) the addressed tasks, (iii) the main techniques used, (iv) the type of analyzed data, and (v) whether a new benchmark dataset was released.

According to the accomplished tasks, existing approaches can be classified as (i) *Date selection* methods, which mainly focused on identifying the key timeline dates, (ii) *Summarization* methods, which particularly address the date summarization step of the TLS pipeline, or (iii) *Full pipeline*, which addressed both the TLS steps. Hereafter, we will separately analyze the previous works belonging to each of the above-mentioned categories.

2.1 Date summarization methods

These works focused on summarizing the news streams relative to long-lasting events by creating a summary per date (both publication and referenced dates were considered). Unlike SDF, they did not select the key dates in the event timeline.

A pioneering work in TLS was presented in [28]. It relied on topic detection and monitoring exploiting entropy-based term evaluation metrics. In [7] the underlying news sub-topics were specified by the end-user through an input query. Hence, TLS was modelled as a sentence-level content retrieval task and solved using ad hoc ranking functions. The use of optimization methods to refine sentence selection both locally at the date level and globally for the whole event timeline was investigated by [35] on news data and by [14, 34] on microblogging data. The current work is not query-driven. To enable timeline updates, it adopts a data-wise approach, i.e., it explores news content separately per date.

[1] https://covidreference.com/timeline (latest access: January 2021)

Paper	Publ. year	Process type	Addressed tasks	Main techniques	Data type	Benchmark release
[28]	2000	Unsupervised	Summarization	KL-based Topic Detection	News	No
[7]	2004	Unsupervised	Summarization	IR-based ranking	News	No
[35]	2011	Unsupervised	Summarization	Local/Global Optimization	News	No
[12]	2012	Unsupervised	Date Selection	Reference and IR-based ranking	News	No
[4]	2013	Supervised	Full Pipeline	Linear Regression	News	No
[20]	2014	Unsupervised	Full Pipeline	Temporal Clustering	News	No
[34]	2015	Unsupervised	Summarization	Incremental clustering	Tweets	No
[31]	2015	Unsupervised	Date Selection	Graph ranking	News	Yes [30] [32]
[18]	2018	Unsupervised	Full Pipeline	Submodular Optimization	News	No
[10]	2020	Supervised	Full Pipeline	Supervised Date Ranking	News	Yes [2]
Our	2021	Unsupervised	Full Pipeline	Graph ranking	News	Yes [3]

Table 1: Literature overview: categorization of existing methods.

2.2 Date selection methods

Few works focused on identifying the most salient dates in the news timeline. Specifically, in [12] the task was accomplished as a re-ranking problem, where the chronological date order was conveniently modified to reflect the number of explicit date references that occurred in the news article corpus.

A more advanced, graph-based ranking strategy was adopted in [31]. The key idea was to model date references using a graph model, where the temporal distance between date pairs was exploited to weigh graph edges. The ranking strategy relied on a influence-based random walk on the graph.

Similar to [31], the date selection process adopted in SDF relies on graph ranking. However, the graph model is enriched with (i) pairwise date similarity measures, which reflect the similarity between the referencing sentence and the summarized content relative to the referenced date and (ii) high-level temporal references, which provide additional knowledge neglected by previous approaches.

2.3 Methods addressing the full pipeline

These works addressed both date selection and summarization tasks. Unlike SDF, all of them prioritize the date selection step in the TLS pipeline. Specifically, the work presented in [4] proposed a supervised date selection method, which ranks the given dates according to the outcomes of a regression model. The goal was to learn a ranking function that embeds the key date relationships. A similar (supervised) strategy was adopted to tackle the date summarization step. An extension of [4] was recently proposed by [10]. They combined a supervised date selection process based on regression models with an unsupervised date summarization step. Unlike [4, 10] the present work proposes to reverse the TLS pipeline (i.e., date summarization first) and relies on a fully unsupervised pipeline. Hence, it can be applied even in the absence of on-topic training data.

To the best of our knowledge, the previous works that are most similar to SDF are [20] and [18]. In the former study, the authors applied a temporal clustering approach on top of an established date ranking strategy [12]. In the latter, the authors proposed a unified optimization-based method, where date selection and summarization constraints were jointly embedded into the objective function. The main drawbacks of the aforesaid methods are (i) the lack of content-based date relevance scores, (ii) the use of date-level temporal references only, and (iii) the need to recompute the summaries as soon as new dates are added to the event timeline. SDF aims at overcoming the above issues.

3 THE PROPOSED PIPELINE

A sketch of the newly proposed pipeline for TimeLine Summarization (TLS), namely *Summarize Dates First* (SDF), is depicted in Figure 1. It consists of the following steps:

(1) **Temporal tagging**: it focuses on extracting temporal references from the news articles' text (see Section 3.1).
(2) **Per-date summary extraction**: it aims at extracting a sentence shortlist separately for each news article corpus published on the same date (see Section 3.2).
(3) **Summary-Driven date selection**: it addresses the selection of the key event dates, together with the corresponding summaries, using both the multiple-level temporal references extracted at Step (1) and the content of the per-date summaries extracted at Step (2) (see Section 3.3).

In the following, we will separately describe each SDF step. Then, in Section 3.4 we will discuss the ability of the SDF system to incrementally update the news timelines as soon as new timestamped data become available.

Outline of the used notation.

- P: period in which the news story happened.
- $\mathbf{A}$: set of news articles published in P and pertinent to the news story.
- $\mathbf{S}$: set of news articles' sentences extracted from $\mathbf{A}$.
- $p(s)$: publication date in P of the article containing sentence $s \in \mathbf{S}$.
- $\mathbf{R}(s)$: set of dates in P referenced by sentence s.
- $\mathbf{R}_{dl}(s)$: subset of dates in $\mathbf{R}(s)$ referenced at the date level.
- $\mathbf{R}_{hl}(s)$: subset of dates in $\mathbf{R}(s)$ referenced at a higher temporal level.
- $\mathbf{C}_d$: news corpus reporting the events occurred on d.

[2] https://github.com/complementizer/news-tls (latest access: April 2021)
[3] https://github.com/MorenoLaQuatra/SDF-TLS

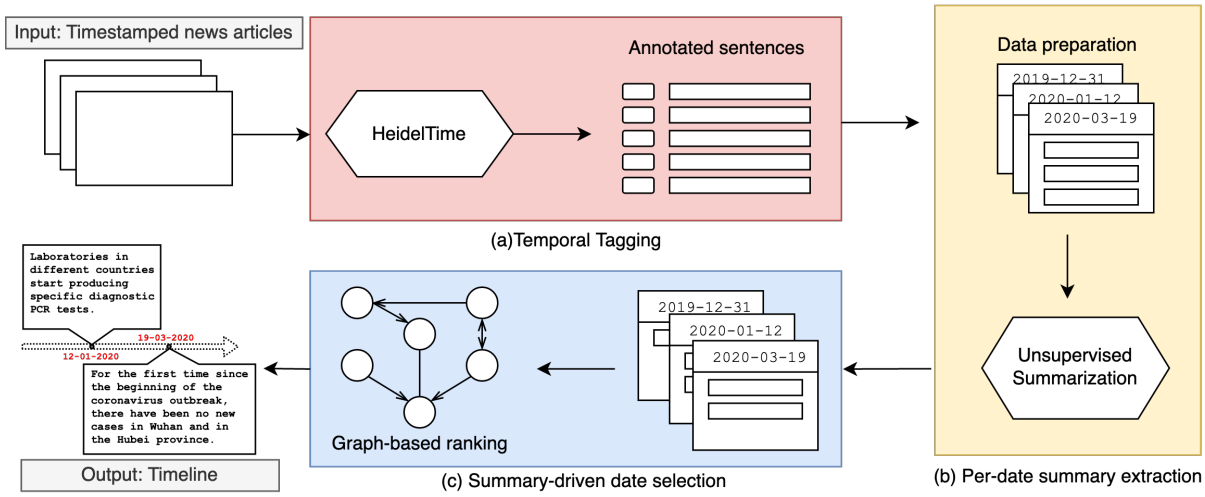

Figure 1: Sketch of the full TLS pipeline.

- *Sum*(C_d): sentence-based summary of the date-specific corpus C_d.

3.1 Temporal tagging

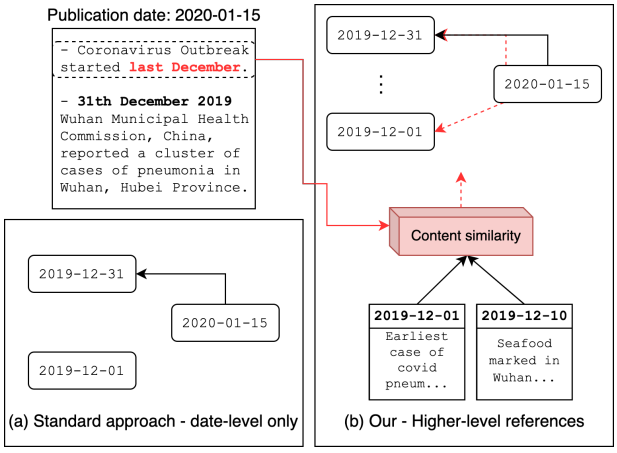

Figure 2: Graph enrichment driven by high-level temporal references and summary content.

This step takes as input the collection of timestamped news articles **A** describing the same topic (e.g., the outbreak and evolution of the Sars-Cov-2 pandemic).

The raw text of the input news articles is first split into sentences using the text parser available in the NLTK library [5] and then annotated using the HeidelTime temporal tagger [27].

A arbitrary sentence $s \in \mathbf{S}$ is annotated with (i) the publication date of the corresponding article, (ii) the date-level references $\mathbf{R}_{dl}(s)$, and (iii) the high-level temporal references $\mathbf{R}_{hl}(s)$.

In the top left corner of Figure 2 an example of annotated news article is depicted. The news article is assigned to a specific news group based on its publication date (2020-01-15). Then, the temporal references $\mathbf{R}(s)$ occurring in the article sentences are exploited to generate date-level references. For instance, the date-level reference *31 December 2019*, written in boldface, links the publication date of the analyzed article to a specific past date in the event timeline. Conversely, the high-level reference *last December*, highlighted in red, indirectly links the article's publication date to all the past dates in December 2019.

The last column of Table 2 reports the percentage of high-level references over the total number of references present in each of the analyzed datasets, i.e.,

$$\sum_{s \in \mathbf{S}} \frac{|\mathbf{R}_{hl}(s)|}{|\mathbf{R}_{dl}(s)| + |\mathbf{R}_{hl}(s)|} \cdot 100$$

Since they all include more than half high-level references, the latter represent a relevant source of knowledge.

The importance of date-level references in TLS was demonstrated by previous research findings (e.g., [18]). Conversely, the information provided by high-level temporal reference was neglected by previous works. Leveraging these particular type of temporal references is one of the main paper goals.

3.2 Per-date summary extraction

It extracts a summary per publication date consisting of the most relevant sentences. Each summary is extracted from a date-specific corpus, which consists of the sentences of all news articles either published on a given date or referencing that particular date are collected.

This step entails a two-step process: firstly, the date-specific corpus is generated. Then, the summarization algorithm is applied.

Data preparation. It takes as input the set **S** of news articles' sentences extracted from the raw articles and produces the date-specific sentence corpus C_d relative to each date d within the reference period P. C_d consists of (i) all the sentences $s \in \mathbf{S}$ such that $p(s)=d$ and (ii) all the sentences $s^* \in \mathbf{S}$ such that $d \in \mathbf{R}_{dl}(s^*)$.

Condition (ii) ensures the inclusion of every sentence that explicitly references a given date (e.g., *31th December 2019 Wuhan Municipal Health Commission, China, reported a cluster of cases...*), as the provided information could be deemed as relevant to extract an informative per-date summary.

Summarization. A summary per date is generated by applying state-of-art unsupervised summarization algorithms. The summarizer generates a summary $Sum(\mathbf{C}_d)$ from each date-specific corpus $\mathbf{C}_d$. The summary consists of a sentence shortlist. According to the official TLS problem statement, the per-date summary length is bounded to N_s sentences [18].

Currently, the SDF method integrates the following sentence-based summarization algorithms:

- TextRank [19]: a traditional graph-based methodology leveraging on syntactic similarity measures.
- LexRank [8]: a standard graph-based approach that relies on the well-known TF-IDF [2] term relevance score.
- CoreRank [29]: a summarizer based on both submodular optimization and graph-based text representation.
- TextRank-BM25 [3]: a variant of the established TextRank summarizer that leverages the Okapi-BM25 [26] score to estimate pairwise sentence similarity.
- ELSA [6]: a recently proposed itemset- and LSA-based summarization system.
- SubModular [16], Centroid-Rank [23], Centroid-Opt [9], EmbeddingRank [22]: four variants of state-of-the-art unsupervised summarizers that leverage on BERT-based contextualized representations [25].

3.3 Summary-driven date selection

This step aims at identifying the key dates in P and returning the corresponding summaries. Unlike previous approaches, SDF considers also per-date summary content $Sum(\mathbf{C}_d)$ and high-level references $\mathbf{R}_{hl}(\cdot)$ in the date selection process.

We present here a new date selection strategy based on graph ranking, namely *Graph-Based Date Selection* (GBDS, in short). It first builds a graph model that embeds the key information provided by both multiple-level date references and summary content. Next, it applies established graph ranking strategies to retrieve the most authoritative dates.

Date graph model. We build a directed graph $G(N, E)$, whose nodes represent distinct publication dates in P, whereas each oriented edge in E connects the node corresponding to a referencing date to the one corresponding to the reference date.

To reward highly cited candidate dates, graph edges are weighted by (i) the date-level reference count and (ii) the pairwise similarity between the content of the sentences including high-level references and that of the referenced date summary. The former contribution rewards the candidate dates that receive many date-level references such as *31th December 2019 Wuhan Municipal Health Commission, China, reported a cluster of cases...*) in Figure 2. In a nutshell, the higher the number of date-level references, the more authoritative the candidate date. The latter contribution weighs the relative importance of a high-level reference based on the summary content. For example, let us the month-level reference such as *Coronavirus outbreak started last December 2019* in Figure 2. It subtends an implicit reward to all publication dates in December 2019. To quantify edge importance we compute the similarity between the citing sentence (i.e. the sentence that includes the high-level reference) and the summary of each publication date in referenced period (December 2019). The idea behind is to reward the dates in December 2019 whose summary content is highly similar to those of the citing sentence, because they are most likely to be actually linked to the referencing date.

Given a monthly period $M \subset P$ referenced by sentence $s \in \mathbf{S}$, the weight of the graph edge $E : p(s) \rightarrow d^r$ s.t. $d^r \in M$ is computed as $sim(Sum(\mathbf{C}_{d^r}), s)$. In the current SDF system implementation, text similarity scores are computed using the n-gram-based ROUGE-2 Precision [15]. To improve the effectiveness of the next graph ranking step, edge weights are first normalized in the range [0,1] and then pruned by filtering out the edges whose weight is below a user-provided cutoff threshold[4].

Graph ranking. To compute the date relevance scores, SDF currently supports the following established ranking functions: (i) Pagerank [22], (ii) HITS [13], (iii) Weighted degree, i.e., the sum of the weights of all incoming and outgoing edges [11], (iv) Weighted in-degree, i.e., i.e., the sum of the weights of all incoming edges [11].

3.4 Incremental timeline updating

The newly proposed variant of the TLS pipeline retrieves the most relevant event dates on top of per-date summarization outcomes. This makes frequent timeline updating more efficient when the updating frequency is rather high and, thus, the number of new dates to be added is limited.

More specifically, as discussed in Section 5.5, the time complexity of the TLS process is mainly influenced by the summarization step, which is approximately one order of magnitude slower than the date selection one. Hence, when the requested frequency of timeline updating is relatively high (e.g., daily or weekly), SDF is more efficient than the traditional pipeline as it requires to generate a significantly lower number of per-date summaries.

Let us assume that the existing event timeline needs to be updated on a daily basis. Every day SDF requires the computation of $n + 1$ per-summary dates, where n is the average number of dates referenced by the new date (e.g, n=5.63 in CovidTLS, n=9.71 in Crisis). Conversely, the traditional pipeline requires to recompute the summaries of *all* the relevant dates, both old and new. For example, on CovidTLS the ground truth comprises 218 dates. Hence, using the traditional pipeline timeline updates would be more computationally expensive.

4 BENCHMARK DATASETS

We carried out experiments on three existing benchmark datasets and a newly released one collecting timestamped news articles relative to the Covid-19 pandemic. Table 2 summarizes the main dataset statistics.

The existing datasets consist of a collection of news documents that ranges over various topics (e.g., 4 topics in Crisis [32], 47 in

[4]We will set the cutoff threshold to 0.8 in our experiments

Entities [10]). Each article in the collection has a publication date. The goal is to separately analyze the articles relative to the same topic in order to extract a separate timeline per topic. For validation purposes, a human-annotated timeline per topic is also given.

The characteristics of the existing datasets are rather diversified. Specifically, they include a rather variable number of articles per topic, ranging from few hundreds (Timeline 17 [30]) to over four thousands (Crisis [32]). The average timeline duration is also quite diversified (in the order of months for Timeline 17 and Crisis, in the order of years for Entities). However, the number of distinct dates per timeline is comparable and not very high (i.e., it varies between 22 and 36). This calls for new benchmark datasets including highly complex topics as well (i.e., many articles and dates).

To tackle the above issue we present CovidTLS, a newly benchmark dataset with peculiar characteristics: just one, complex topic (i.e., the outbreak Covid-19 pandemic) reported by over 26 thousands news articles. The number of candidate dates in CovidTLS is one order of magnitude higher than those of all the existing topics. More details about the analyzed datasets are given below.

The CovidTLS dataset. The newly released CovidTLS dataset describes the outbreak and evolution of the Covid-19 pandemic since the early 2020. Since it has been indisputably among the most relevant worldwide events, it has been reported by an unprecedented amount of news articles. The news article corpus was crawled from several English-written journals. It was annotated with a ground truth timeline retrieved from a public, authoritative website.

CovidTLS is freely available, for research purposes, at https://github.com/MorenoLaQuatra/SDF-TLS.

Other datasets. Timeline 17 [30] comprises 19 news timelines extracted from various news agencies. They ranged over 9 different topics relative to different event types (e.g., catastrophic events or civil wars). Crisis [32] collects event data related to long-term armed conflicts happened in North Africa. Entities [10] contains timeline data for entities rather than events. Specifically, it consists of 47 different timelines ranging over an equal number of topics. Most of the covered topics are related to life-spanning events of famous people. The remaining ones are related to business companies and no-profit organizations.

5 EXPERIMENTS

We performed an extensive experimental validation of the performance of the SDF approach. The experiments were run on a machine equipped with Intel® Xeon® Gold 5115 CPU, 512 GB of RAM and running Ubuntu 18.04.5 LTS.

Hereafter, we will separately discuss results achieved on the date selection and summarization steps. More details on the evaluation process are given below.

Date selection evaluation. The date selection problem can be reformulated as the following Information Retrieval task: given a collection of timestamped news articles relative to a specific news topic, retrieve the dates that are most relevant to the news story according to the reference timeline (i.e., the ground truth).

For each dataset and system, TLS performance was measured in terms of the average F1-score over all the analyzed topics, which is the harmonic mean of retrieval precision and recall [24].

Date summarization evaluation. Text summarization outcomes are commonly evaluated using the established Rouge toolkit [15]. It counts the unit overlaps between an automatically generated summary and a reference summary (i.e., the ground truth), which is typically hand-written by domain experts. Depending on the type of considered textual unit, different Rouge metrics can be analyzed (e.g., Rouge-1 for unigrams, Rouge-2 for bigrams). Separately for each rouge metric TLS system performance is quantified by the corresponding precision, recall, and F1-score values [24].

In the TLS scenario both source data and summary content are timestamped. Hence, Rouge-based summary evaluation is tailored to TLS by considering the timeliness of the selected content. Specifically, we considered the following established Rouge variants: *concatenation*, *agreement*, and *alignment* [17]. *Concatenation-based scores* (*concat*, in short) replicate the standard ROUGE evaluation by merging all the separate per-date summaries into a unique summary, regardless of the associated timestamp. *Agreement-based scores* (namely *agreement*) limit the summary comparisons to the dates that actually occur in the ground truth. *Alignment-based scores* (namely *align*) rely on the following steps:

(1) Align the dates of the extracted and reference timelines using a many-to-one mapping function, which firstly looks for an exact date matching. Whenever it is not found, a approximated matching is generated by retrieving to the closest date in the reference timeline.
(2) Compute the average ROUGE scores, according to the aligned timeline versions.

Tested methods. We compared the results achieved by the SDF system with that of the following approaches.

- The IR-based approach proposed in [7].
- The system proposed in [18], which is, to the best of our knowledge, the best performing state-of-the-art *unsupervised* TLS method.
- The regression-based approach proposed in [10], which is (to the best of our knowledge) the latest *supervised* TLS approach.

Separately for each dataset and method, the best TLS performance were achieved by exploring multiple configuration settings via grid search.

5.1 Date summarization results

Table 3 reports the Rouge-1 and Rouge-2 F1-scores achieved by all the tested TLS methods on the date summarization step. Although all summarization outcomes are comparable with each other, we graphically separate unsupervised methods, like SDF, from the supervised one [10], as they are conceptually different. For the sake of readability, the results of the best performing unsupervised method are written in boldface. For each SDF competitor and dataset, we report here the best result provided by the respective authors. For SDF the best configuration setting is specified in brackets.

On Timeline 17, Crisis, and Entities datasets we assessed also the statistical significance of the difference in performance between two

Data collection	# topics	Avg.# articles per topic	Avg. # dates in timeline	Avg. timeline duration	% of high-level references
Timeline 17	9	513.11	36.42	242.47 days	50.99
Crisis	4	4,560.75	29.22	387.86 days	54.61
Entities	47	1,086.49	22.57	19.02 years	62.68
CovidTLS (our)	1	26,376	218	266 days	61.07

Table 2: characteristics of the benchmark datasets.

TLS systems using the two-sided paired approximate randomization test [21][5]. On CovidTLS the statistical test was not applicable as it focuses on a single topic. Every statistically significant performance worsening against the best performing method is starred in Table 3.

SDF performed averagely best against all the tested unsupervised methods. On Timeline 17 the Rouge scores were comparable to those of the state-of-the-art supervised one (i.e., DateWise [10]), despite the latter was facilitated by the a priori knowledge of training data. On CovidTLS, the SDF performance was significantly better than all other methods: the achieved scores were roughly ten times higher than the state-of-the-art unsupervised method whereas doubled the supervised one.

5.2 Date selection evaluation

Table 4 summarizes the best results achieved for the date selection task, where we separate again unsupervised methods from the supervised one. To explore the effect of the graph ranking method on SDF performance, we report the results achieved using different strategies.

SDF performed best on all the tested dataset. On 3 out of 4 datasets, its performance was superior to that of the state-of-the-art supervised method as well.

5.3 Effect of summarized knowledge on date selection

We explored the impact of using summarized content, extracted from high-level references, on TLS performance. The goal is to understand whether and to what extent summary-driven content analyses are useful for tackling the date selection task.

Figure 3 shows a result comparison, in terms of minimum, average and maximum F1-scores, between the graph-based strategies relying on Summary-Driven Content-Based analysis (SDCB) and not (no-SDCB).

The results show that integrating summary content in date selection was particularly effective when the article news flow is likely to be redundant. Specifically, it has shown to be helpful while coping with (i) long-lasting news topics (e.g., Entities) or (ii) complex, multi-faceted topics (e.g., CovidTLS). Conversely, its use was detrimental on smaller news datasets (e.g., Timeline 17), where summarizing per-date news content was apparently not beneficial.

[5]To run the tests we used the Python-based implementation publicly available at https://github.com/smartschat/art (latest access: January 2021)

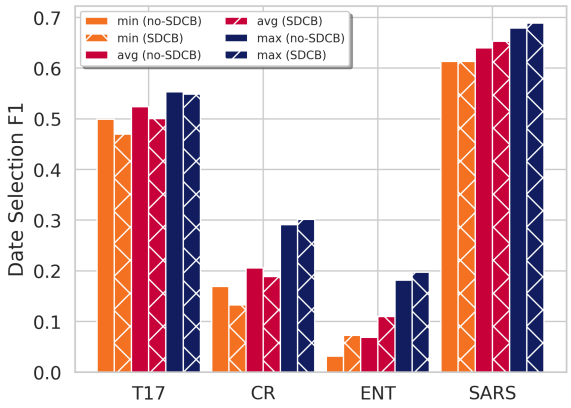

Figure 3: Comparison between content-based (SDCB) and content-independent (no-SDCB) approaches.

5.4 Comparison between summarization algorithms

We compared also the performance of different text summarization algorithms independently of the date selection strategy, which was selected according to the results summarized in Section 5.2 and kept fixed.

Table 5 reports the results achieved using nine different summarization algorithms. The results show that the graph-based approach presented in [3] averagely performed best on the analyzed datasets. The achieved results are in line with the previous research findings [31], as graph ranking methods have shown to be particularly effective in coping with timestamped news datasets.

5.5 Time complexity

We empirically analyzed the time complexity of the SDF approach. Regarding the date selection step, the times spent in date selection per timeline without considering news content (i.e., no-SDCB strategy) ranged from 7s (Timeline 17) to 134s (CovidTLS). By integrating the summarized knowledge, the computational times yielded 3x-5x increase and ranged from 19s to 726s.

The time complexity of the date summarization step depends on the selected algorithm. For example, by focusing on the best performing summarizer, i.e., TextRank-BM25 [3], the execution times varied between 65.19s (Entities) to 1032.43s (CovidTLS). In conclusion, the summarization step was approximately one order of magnitude more computationally expensive than the date selection step.

Model	Type	concat		agreement		alignment-m21	
		R1-F1	R2-F1	R1-F1	R2-F1	R1-F1	R2-F1
Timeline 17							
Chieu & Lee [7]	U	0.275*	0.065*	0.028*	0.008*	0.057*	0.014*
TLS+reweighting [18]	U	0.383	0.092	0.094	0.025*	0.109	0.028
$SDF_{\text{degree, TextRank-BM25}}$	U	**0.401**	**0.101**	**0.106**	**0.033**	**0.120**	**0.035**
DateWise [10]	S-DS	0.385	0.097	0.107	0.032	0.120	0.035
Crisis							
Chieu & Lee [7]	U	**0.368**	0.066	0.028*	0.005*	0.051*	0.009*
ASMDS+DateRef [18]	U	0.333*	0.07	0.051	0.011	0.073	0.016
$SDF_{\text{in-degree-SDCB, TextRank-BM25}}$	U	0.360	**0.073**	**0.064**	**0.014**	**0.086**	**0.018**
DateWise [10]	S-DS	0.347	0.075	0.071	0.023	0.089	0.026
Entities							
Chieu & Lee [7]	U	**0.275**	**0.053**	0.025*	0.011	0.038*	0.012
TLS+reweighting+DateRef [18]	U	**0.275**	**0.053**	0.039	**0.013**	**0.051**	**0.015**
$SDF_{\text{in-degree-SDCB, TextRank-BM25}}$	U	**0.275**	0.052	**0.041**	0.011	**0.051**	0.014
DateWise [10]	S-DS	0.271	0.051	0.045	0.014	0.057	0.017
CovidTLS							
Chieu & Lee [7]	U	0.203	0.021	0.008	0.001	0.017	0.001
ASMDS+TempDiv+DateRef [18]	U	0.249	0.036	0.028	0.001	0.03	0.001
$SDF_{\text{Pagerank-SDCB, TextRank-BM25}}$	U	**0.439**	**0.076**	**0.062**	**0.011**	**0.072**	**0.012**
DateWise [10]	S-DS	0.318	0.038	0.036	0.005	0.040	0.006

Table 3: Date summarization: Rouge-based evaluation.

Model		Type	Timeline 17	Crisis	Entities	CovidTLS
Chieu & Lee [7]		U	0.230*	0.166*	0.09*	0.176
Martschat & Markert [18]	ASMDS-based	U	0.531	0.278	0.163	0.685
	TLS-constraint-based	U	0.527	0.266	0.180	0.679
SDF	Weighted in-degree	U	0.549	**0.302**	**0.197**	**0.689**
	HITS	U	**0.553**	0.206	0.095	0.679
	Pagerank	U	0.537	0.175	0.161	0.623
	Weighted degree	U	0.532	0.275	0.117	0.679
Ghalandari & Ifrim [10]		S	0.544	0.295	0.205	0.679

Table 4: Date selection: IR-based evaluation in terms of F1-score.

6 CONCLUSIONS AND FUTURE WORK

We propose a new strategy, called SDF, to reverse the mainstream TLS pipeline. In a nutshell, SDF works as follows: *It summarizes dates first. Next, it selects the key dates by conveniently exploiting both multiple-level temporal references and the previously summarized knowledge*. This approach allows us to overcome the limitations of previous date-wise approaches, which entail (i) exploring the content of textual news articles to drive the date selection process, (ii) taking advantage of the presence of high-level temporal references, and (iii) incrementally updating the news timelines as soon as new candidate dates are added to the raw news collections.

A thorough experimental evaluation shows that SDF on average performed best against all the previous unsupervised approaches, whereas was competitive against supervised methods. This makes SDF particularly appealing when there is a lack of on-topic training news data (see Section 5.1). The experimental analysis yielded the following takeaways.

- The summarized knowledge extracted from high-level references is useful to identify the key event dates when either the news story is long-lasting or the covered topic is rather complex and multi-faceted (see Section 5.3).
- Real news collections include a large number of high-level temporal references that can be profitably exploited in TLS (see Table 2).
- Graph-based approaches appeared to be the most effective strategies to summarize per-date news articles (see Section 5.4).

As future work, we plan to extend the current strategy to handle cross-lingual and multimodal collections. The former task entails extracting relevant content from streams of news articles written in different languages. The latter focuses on generating news timelines including also images, videos, and social content [33]. Furthermore,

Summarizer	concat F1		agreement F1		align+m:1 F1	
	R1	R2	R1	R2	R1	R2
Timeline 17						
TextRank	0.363*	0.084*	0.086*	0.023*	0.097*	0.025*
LexRank	0.370*	0.084*	0.088*	0.025*	0.100*	0.027*
CoreRank	0.371*	0.091*	0.092*	0.024*	0.105*	0.026*
TextRank-BM25	**0.401**	**0.101**	**0.106**	**0.033**	**0.120**	**0.035**
ELSA	0.389*	0.097	0.100	0.029	0.114	0.032
SubModular	0.367*	0.082*	0.086*	0.024	0.098*	0.025
Centroid-Rank	0.365*	0.082*	0.084*	0.023	0.096*	0.025
Centroid-Opt	0.372*	0.082*	0.084*	0.021*	0.097*	0.023*
EmbeddingRank	0.365*	0.084*	0.087*	0.022	0.098*	0.024*
Crisis						
TextRank	0.311*	0.058*	0.043*	0.009	0.062*	0.012
LexRank	0.312*	0.056*	0.042*	0.009*	0.059*	0.012*
CoreRank	0.356	**0.075**	0.060	0.014	0.080	0.017
TextRank-BM25	**0.360**	0.073	**0.064**	0.014	**0.086**	**0.018**
ELSA	0.338*	0.064*	0.061	**0.015**	0.081	**0.018**
SubModular	0.337	0.057*	0.050*	0.009	0.068*	0.012*
Centroid-Rank	0.335*	0.056*	0.047*	0.008*	0.065*	0.011*
Centroid-Opt	0.337*	0.057*	0.049*	0.009*	0.068*	0.012*
EmbeddingRank	0.337	0.056*	0.048*	0.008*	0.066*	0.011*
Entities						
TextRank	0.238*	0.041*	0.030*	0.007*	0.040*	0.010*
LexRank	0.245*	0.043*	0.032*	0.008*	0.041*	0.010*
CoreRank	0.258*	0.049	0.038	**0.012**	0.048	**0.014**
TextRank-BM25	**0.275**	**0.052**	**0.041**	0.011	**0.051**	**0.014**
ELSA	0.258*	0.044*	0.036*	0.009	0.046*	0.011
SubModular	0.249*	0.040*	0.031*	0.007*	0.040*	0.009*
Centroid-Rank	0.251*	0.042*	0.032*	0.008*	0.041*	0.009*
Centroid-Opt	0.251*	0.041*	0.032*	0.007*	0.041*	0.009*
EmbeddingRank	0.250*	0.041*	0.032*	0.007*	0.041*	0.009*
CovidTLS						
TextRank	0.451	0.061	0.045	0.004	0.054	0.004
LexRank	**0.461**	0.065	0.051	0.005	0.061	0.006
CoreRank	0.383	0.053	0.044	0.003	0.051	0.004
TextRank-BM25	0.439	**0.076**	**0.062**	**0.011**	**0.072**	**0.012**
ELSA	0.428	0.065	0.050	0.005	0.058	0.005
SubModular	0.424	0.055	0.051	0.006	0.059	0.006
Centroid-Rank	0.419	0.053	0.050	0.005	0.058	0.006
Centroid-Opt	0.433	0.057	0.049	0.005	0.057	0.006
EmbeddingRank	0.425	0.057	0.050	0.006	0.058	0.007

Table 5: Comparison between summarization algorithms.

we plan to conduct a qualitative user study to assess summary informativeness, coverage, and diversity.

ACKNOWLEDGMENTS

The research leading to these results has been co-funded by Radiotelevisione Italiana (RAI), Centre for Research and Technological Innovation and the SmartData@PoliTO center for Big Data and Machine Learning technologies.

REFERENCES

[1] Javed A. Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. TREC 2015 Temporal Summarization Track Overview. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015 (NIST Special Publication, Vol. 500-319)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). http://trec.nist.gov/pubs/trec24/papers/Overview-TS.pdf

[2] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., USA.

[3] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606* (2016).

[4] Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd International Conference on World Wide Web*. 91–92.

[5] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[6] Luca Cagliero, Paolo Garza, and Elena Baralis. 2019. ELSA: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–33.

[7] Hai Leong Chieu and Yoong Keok Lee. 2004. Query Based Event Extraction along a Timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sheffield, United Kingdom) *(SIGIR '04)*. Association for Computing Machinery, New York, NY, USA, 425–432. https://doi.org/10.1145/1008992.1009065

[8] Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.

[9] Demian Gholipour Ghalandari. 2017. Revisiting the Centroid-based Method: A Strong Baseline for Multi-Document Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, Copenhagen, Denmark, 85–90. https://doi.org/10.18653/v1/W17-4511

[10] Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the State-of-the-Art in News Timeline Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1322–1334. https://doi.org/10.18653/v1/2020.acl-main.122

[11] Ismael A Jannoud and Mohammad Z Masoud. 2015. On understanding centrality in directed citation graph. In *Advanced Computer and Communication Engineering Technology*. Springer, 43–51.

[12] Rémy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. 2012. Finding Salient Dates for Building Thematic Timelines. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jeju Island, Korea, 730–739. https://www.aclweb.org/anthology/P12-1077

[13] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.

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

[15] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://www.aclweb.org/anthology/W04-1013

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

[17] Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for Timeline Summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 285–290. https://www.aclweb.org/anthology/E17-2046

[18] Sebastian Martschat and Katja Markert. 2018. A Temporally Sensitive Submodularity Framework for Timeline Summarization. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Brussels, Belgium, 230–240. https://doi.org/10.18653/v1/K18-1023

[19] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[20] Kiem-Hieu Nguyen, Xavier Tannier, and Veronique Moriceau. 2014. Ranking Multidocument Event Descriptions for Building Thematic Timelines. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 1208–1217. https://www.aclweb.org/anthology/C14-1114

[21] Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses.* Wiley New York.

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

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

[24] Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets.* Cambridge University Press, New York, NY, USA.

[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[26] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[27] Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation* 47, 2 (2013), 269–298. https://doi.org/10.1007/s10579-012-9179-y

[28] Russell Swan and James Allan. 2000. Automatic Generation of Overview Timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece) *(SIGIR '00).* Association for Computing Machinery, New York, NY, USA, 49–56. https://doi.org/10.1145/345508.345546

[29] Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the workshop on new frontiers in summarization.* 48–58.

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

[31] Giang Tran, Eelco Herder, and Katja Markert. 2015. Joint Graphical Models for Date Selection in Timeline Summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* Association for Computational Linguistics, Beijing, China, 1598–1607. https://doi.org/10.3115/v1/P15-1154

[32] 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 2013 Workshop on Time-aware Information Access (TAIA.*

[33] 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 *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, San Diego, California, 58–68. https://doi.org/10.18653/v1/N16-1008

[34] Zhenhua Wang, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2014. On summarization and timeline generation for evolutionary tweet streams. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2014), 1301–1315.

[35] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Beijing, China) *(SIGIR '11).* Association for Computing Machinery, New York, NY, USA, 745–754. https://doi.org/10.1145/2009916.2010016