Archive TimeLine Summarization (ATLS): Conceptual Framework for Timeline Generation over Historical Document Collections

Nicolas Gutehrlé
Laboratoire CRIT,
University of Bourgogne
Franche-Comté, France
nicolas.gutehrle
@univ-fcomte.fr

Antoine Doucet
Laboratoire L3i,
University of La Rochelle,
France
antoine.doucet
@univ-lr.fr

Adam Jatowt
Dept. of Computer Science &
Digital Science Center,
University of Innsbruck, Austria
adam.jatowt
@uibk.ac.at

Abstract

Archive collections are nowadays mostly available through search engines interfaces, which allow a user to retrieve documents by issuing queries. The study of these collections may be, however, impaired by some aspects of search engines, such as the overwhelming number of documents returned or the lack of contextual knowledge provided. New methods that could work independently or in combination with search engines are then required to access these collections. In this position paper, we propose to extend *TimeLine Summarization* (TLS) methods on archive collections to assist in their studies. We provide an overview of existing TLS methods and we describe a conceptual framework for an *Archive TimeLine Summarization* (ATLS) system, which aims to generate informative, readable and interpretable timelines.

1 Introduction

1.1 Exploring archives

In the recent years, archives and libraries across the world have frequently conducted digitization campaigns of their collections. This first opened access to thousands of historical documents to a wider public, but also propelled the emergence of new research fields such as Digital Humanities and Digital History. These collections are usually accessible through search engines, which return documents relevant to a query specified by the user. Unfortunately, standard search engines are not fully suited to assist users in exploring historical collections such as news archives where temporal aspects of documents play a key role. Firstly, search engines return documents by their relevance to the query, typically without considering the chronological or causal relations between them, which may prevent the user from understanding the interrelations between events. Furthermore when exploring such documents, the user might lack the contextual knowledge to understand the events that are mentioned in them. This is especially true when exploring news archives coming from distant pasts or exploring longitudinal collections, i.e. which span over a long time frame such as decades or centuries. Search engines do not seem to consider the importance of an event mention for a given query, thus less important events might be returned by the system, especially for broad queries. Improved search engines are then required to study such collections.

1.2 Augmenting search engines with timelines

One promising method to improve the output of search engines operating over archival collection is *TimeLine Summarization* (TLS). TLS consists in summarizing multiple documents by generating a timeline where important events detected in the dataset are associated with a time unit such as a day. TLS is a subfield of the *Multi-Document Summarization* (MDS) task and has been studied extensively in the NLP community: for instance, Swan and Allan (2000) generate clusters of Named Entities and noun chunks that best describe major news topics covered in a subset of the TDT-2 dataset (Allan et al., 1998), which contains text transcripts of broadcast news spanning from January 1, 1998, to June 30, 1998, in English; Nguyen et al. (2014) generate timelines by detecting events that are the most relevant to a user query. They apply their methodology on a dataset of newswire texts in English covering the 2004-2011 period provided by the AFP French news agency; Duan et al. (2017) extend these methods to summarize the common history of similar entities such as Japanese Cities or French scientists. Examples of timelines generated by such methods are shown in Figure 1.

Hence, TLS could serve as a distant reading tool and as a first step in exploring a dataset by providing an overview of its key events. Moreover, TLS could be combined with search engines and used as an interface to search results returned by issuing queries over large datasets, as suggested in Swan and Allan (2000); Alonso et al. (2021). From there,

Proceedings of the 6th LaTeCH-CLfL Workshop, pages 13–23
October 16, 2022. ©2022 International Conference on Computational Linguistics

Date	Summary
2011-01-25	Thousands of protesters spilled into the streets of Egypt on Tuesday , an unprecedented display of anti-government rage inspired in part by the tumult in the nearby North African nation of Tunisia.
2011-01-26	Twitter says its site is being blocked in Egypt Egyptians took to the streets in what could be a sequel to the recent revolution in Tunisia witter , Facebook and YouTube were widely used in Tunisia 's uprising and in Iran last year -LRB-.
2011-01-28	With parts of his capital ablaze , Mubarak said he was asking his government to resign and would soon announce a new one , pledging to address the concerns of thousand of Egyptians protesting in Cairo 's streets . Amre Moussa , the Arab League 's secretary-general and a veteran Egyptian diplomat , joined protesters in Cairo 's Tahrir Square on Friday , state-run Nile TV reported .

From	To	Top headlines
5/2010	7/2011	Syrian officials launch tear gas against protesters Security forces shoot at protesters New York Times journalist with the Pulitzer died of an Asthma attack in Syria
8/2011	3/2012	Assad promises elections in February in Syria US withdraws ambassador from Syria for security reasons NATO says goodbye to Libya and the world turns to Syria
7/2012	12/2012	Meeting of senior officials in Geneva failed agreement to end violence in Syria Russia delivers three war helicopters to Syria Red Cross says Syria is in civil war
7/2016	11/2016	Maternity unit among hospitals bombed in Idlib air strikes Russian helicopter shot down in Syria. Turkish army enters Syria

Figure 1: Examples of generated timelines by Yu et al. (2021) (left) and Campos et al. (2018) (right), summarizing a set of documents about respectively Egyptian protests and the Syrian War. The left timeline outputs a summary on a day-to-day basis, whereas the right timeline lists events using uneven periods of time.

the user could zoom into the documents in order to proceed to close reading. Furthermore, these summaries would be presented in chronological order, thus preserving the link between events, and could also be contextualized by adding data from external knowledge bases as in Ceroni et al. (2014).

Search engines augmented with timelines would be especially useful in a Digital Humanities (DH) context such as for facilitating the study of historical datasets, as they would provide necessary context to understand past events and to structure the event landscape. They could also help the user understand the history of a particular entity such as a person or a location, or even a group of such entities through providing a bird's-eye view of the relevant data. A good example of such search engine augmented with TLS is the Conta-me Histórias (Tell me stories) platform[1], where the user can query news articles from the Portuguese web archive. The user-friendly interface allows a distant reading of the documents returned by the query through a timeline that summarizes them, but also allows close reading by preserving the link to the original documents. To the best of our knowledge, works on applying TLS methods to structure archives of historical documents, or more broadly in the Digital Humanities field, are quite scarce.

1.3 Challenges of applying TLS to archives

Unfortunately, several aspects of such archives make the application of TLS methods not straightforward: first, these datasets are often processed with Optical Character Recognition (OCR). Previous studies have shown that downstream tasks such as Named Entity Recognition (NER), Event Detection (ED) (Boros et al., 2022), Topic Modelling (TM) (Mutuvi et al., 2018) or Named Entity Linking (NEL) (Linhares Pontes et al., 2019) are impacted by the quality of the OCR output.

[1] https://contamehistorias.pt/arquivopt

To our knowledge, there is no study on the impact of OCR on TLS, but we can assume it will be similar. Furthermore, archive collections may also differ from contemporary data because of their temporal context: orthographic rules may differ, places might have changed names (Smith and Crane, 2001) or concepts may have acquired another meaning. Most existing annotated resources necessary for NLP components such as NER or ED are created on contemporary data. Historical documents archives are thus harder to process because of this lack of suitable annotated resources.

Most TLS methods generate timelines through statistical analysis of the input dataset. They also often require that the input corpus contains documents of a similar type and similar content. However, an archive collection may be heterogeneous and contain documents of different authors, genres, topics and periods. It may also be fragmentary and not as complete as a contemporary dataset. Finally, although the timelines generated by TLS systems are often easy to read, the process that created them is often not made explicit. If timelines must assist the study of historical datasets by highlighting important events, they must be interpretable and explain why these events are deemed important.

In this position paper, we propose to extend *TimeLine Summarization* (TLS) methods to assist in the studies of archive collections. We first present an overview of existing TLS methods. We then describe a conceptual framework for an *Archive TimeLine Summarization* (ATLS) system, which aims to generate informative, readable and interpretable timelines, before suggesting several methods to implement it.

This paper is organized as follows: in Section 2 we present an overview of existing TimeLine Summarization methods. In Section 3 and 4, we respectively describe our conceptual framework and discuss some of its potential applications. Finally, we present our conclusion in Section 5, alongside possibilities for future works.

2 Related Work

2.1 TimeLine Summarization

Most TLS methods generate timelines by applying the two following steps: the **Date Selection** step which identifies and ranks the key dates in the documents, and the **Date Summarization** step which generates a summary of an event occurring at a specific date by picking important sentences in the documents published on that date. To identify important dates in the dataset, Gholipour Ghalandari and Ifrim (2020) select the most frequent date mentions, Tran et al. (2015b) use a graph-ranking model and Kessler et al. (2012) combine a clustering model and a supervised classifier. For the second step, La Quatra et al. (2021) apply state-of-the-art methods for Text Summarization (TS) such as TextRank (Mihalcea and Tarau, 2004) whereas Martschat and Markert (2018) adapt methods from the Multi-Document Summarization (MDS) field. TLS has been generally extractive, i.e. the summary is created by copying textual elements (e.g., sentences or paragraphs) from the input data (Tran et al., 2015a). Other works are abstractive, i.e. the summary is a completely new text generated by the system (Steen and Markert, 2019).

TLS methods in general tend to be applied to summarize datasets describing large events, such as the Egyptian protests or the Syrian War (Tran et al., 2015b; Martschat and Markert, 2018). These methods require that the dataset covers a constrained period of time and is homogeneous, i.e. that the documents cover the same topic. Standard TLS methods are thus not suited to summarize heterogeneous or longitudinal datasets. Some works such as Nguyen et al. (2014); Kessler et al. (2012); Chieu and Lee (2004); Pasquali et al. (2019) can be described as *Query-based TimeLine Summarization* (QTLS), as they apply TLS on documents related to a user query such as documents returned by a search engine.

QTLS generally consists in the two following steps: *Event Detection* and *Event Ranking*. To detect events, Chieu and Lee (2004) select any sentence where the terms of the query appear, Nguyen et al. (2014) cluster by a common date every sentence returned by the query and Pasquali et al. (2019) detect peaks of date occurrences in the time span covered by the documents. Other works train a classifier to detect important events (Chasin, 2010) or rank events by their importance with a Learning-to-Rank model (Ge et al., 2015). However, these classifiers need training data, which are difficult to create since defining what is important is a subjective matter. This can lead to disappointing results as shown in Chasin (2010). To determine the importance of events, Nguyen et al. (2014) first score them according to their relevancy and saliency to the query, then rerank them to ensure a diverse timeline. Chieu and Lee (2004) rank the importance of a sentence according to their "interest" and "burstiness", then remove duplicate sentences to ensure diversity. Pasquali et al. (2019) use the keyword extractor YAKE! (Campos et al., 2018) to weight the terms in the event description. Duplicate event descriptions are detected with the Levenshtein similarity measure and removed. Those methods finally select the top most important events to generate the timeline.

In order to generalize the application of TLS, Yu et al. (2021) propose a Multiple TimeLine Summarization (MTLS) system, which generates a timeline for each story found in the dataset. To do so, it first detects events mentioned in the dataset and measures their saliency and consistency. An event linking step determines the link between these events in order to generate each timeline. Similarly, Duan et al. (2020) propose the *Comparative TimeLine Summarization* (CTLS) task, which generates a comparative timeline highlighting the contrast between two timestamped timeline documents (e.g. biographies, historical sections, ...) by computing local and global importance of events.

There are few datasets for the TLS task such as 17 Timelines (T17) (Tran et al., 2013), CRISIS (Tran et al., 2015a), ENTITIES (Gholipour Ghalandari and Ifrim, 2020), CovidTLS (La Quatra et al., 2021) or TLS-Covid19 (Pasquali et al., 2021) which are constructed from contemporary news articles. However, datasets are often lacking in most projects. It is then necessary to create a dataset from scratch as in Minard et al. (2015); Nguyen et al. (2014); Ge et al. (2015); Bedi et al. (2017) or extend existing ones as in Yu et al. (2021).

Due to this lack of datasets, evaluating TLS systems is a difficult task. The date selection step can be evaluated with the F1-measure (La Quatra et al., 2021; Gholipour Ghalandari and Ifrim, 2020) or with the Mean Average Precision (MAP) metric (Nguyen et al., 2014). The date summary is often evaluated with one of the ROUGE metrics (Lin,

2004) to compare a ground-truth timeline and a generated one (Nguyen et al., 2014; Duan et al., 2020; Yu et al., 2021; Gholipour Ghalandari and Ifrim, 2020). Methods relying on event detection such as Ge et al. (2015); Minard et al. (2015); Bedi et al. (2017) often evaluate their system in terms of Precision, Recall and F1-measure. However, most projects often lack datasets and must then resort to human evaluation as in Duan et al. (2017); Swan and Allan (2000); Tran et al. (2015a).

2.2 TLS Variants

We present below formal definitions of several existing TLS variants:

TLS: takes as input a standalone homogeneous dataset of timestamped documents $\mathcal{D} = \{d_1, d_2, ..., d_{|D|}\}$ and generates a timeline $T = \{p_1, p_2, ..., p_{|T|}\}$ of time-summary pairs $p_i = (t_i, s_i)$, where s_i summarizes important events happening at time t_i;

QTLS: outputs a timeline $T = \{p_1, p_2, ..., p_{|T|}\}$ as a sequence of time-summary pairs $p_i = (t_i, s_i)$ from a set of timestamped documents $\{d_1, d_2, ..., d_{|D|}\}$ based on a query $\mathcal{Q} = \{w_1, w_2, ..., w_k\}$ where w_i denotes a word belonging to the query;

MTLS: takes as input a dataset of timestamped documents $\mathcal{D} = \{d_1, d_2, ..., d_{|D|}\}$ that can be standalone or returned using a query $\mathcal{Q} = \{w_1, w_2, ..., w_k\}$, and outputs a set of timelines $\mathcal{T} = \{T_1, T_2, ..., T_m\}$ for each story or topic detected in $\mathcal{D}$, where each timeline T_i is a sequence of time-summary pairs $p_i = (t_i, s_i)$;

CTLS: takes as input two datasets of timestamped documents $\mathcal{D}_\mathcal{A} = \{d_1, d_2, ..., d_{|D_A|}\}$ and $\mathcal{D}_\mathcal{B} = \{d_1, d_2, ..., d_{|D_B|}\}$ and outputs two timelines $\mathcal{T}_\mathcal{A}$ and $\mathcal{T}_\mathcal{B}$ made of contrasting events detected in $\mathcal{D}_\mathcal{A}$ and $\mathcal{D}_\mathcal{B}$, each as a sequence of time-summary pairs $p_i = (t_i, s_i)$;

3 Framework

In this section, we present a conceptual framework for an *Archive TimeLine Summarization* (ATLS) which addresses the challenges raised by archive collections such as the sparsity of data, OCR problems, context shifts and linguistic changes over time in order to generate timelines based on these datasets. We first provide a definition of ATLS and describe the type of dataset expected before presenting the framework and discussing how to evaluate its output.

3.1 Overview

The framework consists of the two key steps: *Timeline Generation* and *Timeline Presentation*. The first step extracts textual elements describing an event and attributes them an importance score. The second one generates the timeline by filtering events and selecting their description.

The processing stages of the framework are shown in Figure 2. The first step has to run only once over the processed dataset, since it aims to detect the elements composing the timeline to be generated. In contrast, the second step can be run multiple times to update the timeline.

3.2 Problem Definition

We define ATLS as follows:

Input: A longitudinal dataset of timestamped documents $\mathcal{D} = \{d_1, d_2, ..., d_{|D|}\}$ taken from an archival collection, either standalone or returned by a query $\mathcal{Q} = \{w_1, w_2, ..., w_k\}$. The period of time covered by $\mathcal{D}$ is usually much longer than the one typically used in TLS.

Output: A timeline $\mathcal{T}$ generated from $\mathcal{D}$ as a sequence of time-summary pairs $p_i = (t_i, s_i)$, where s_i summarizes important events happening at time t_i.

We compare the key characteristics of TLS and ATLS in Tab. 1.

3.3 Expected Dataset

The framework takes as input a longitudinal dataset composed of timestamped documents, such as news articles from a historical newspaper collection. This dataset can be standalone or made of documents returned by a search engine for a given query $\mathcal{Q}$. The dataset could be in raw format or have been pre-processed. We would suggest at least the two following pre-processing steps: first, we recommend to clean the dataset if it has been processed with OCR, either manually or semi-automatically, since the OCR quality will impact further steps (Nguyen et al., 2021). Secondly, we recommend to detect temporal expressions, as they are a good indicator of event mentions. Temporal expressions are either explicit (e.g. February 17, 1995) or implicit (e.g. yesterday, next month). One can use tools such as HeidelTime (Strötgen and Gertz, 2010) or SUTime (Chang and Manning,

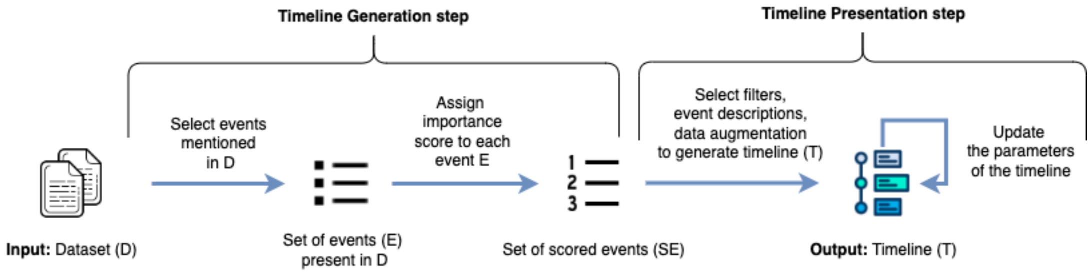

Figure 2: Conceptual pipeline for building the ATLS system

2012) to detect temporal expressions in text and resolve them to an absolute date format, simplifying their use in the TLS process. However, we must keep in mind that the detection of temporal expressions, especially implicit ones, is still a challenging task. Moreover, available tools such as these were mainly conceived for contemporary data, and thus may not work as properly on historical data.

The input dataset could be pre-processed further by applying NLP components such as Name Entity Recognition (NER), Topic Modelling (TM), Event Extraction (EE), Relation Extraction (RE), Keyword Extraction (KE), or Keyword Generation (KG). Such annotations could be used to index the dataset and allow the user to query documents about a specific Named Entity or topic, as in the impresso[2] or the NewsEye[3] platforms.

3.4 Timeline Generation

In this section, we present the first main step of the framework, which extracts mentions of events and attributes them an importance score.

3.4.1 Event Detection

Although events can be defined in many ways, a commonly accepted definition is "something that is *happening* or that is holding true in a given circumstance", as stated in the TimeML guidelines Saurí et al. (2006). Events can be detected in multiple ways: one could detect them through statistical analysis of the corpus. For instance, Chieu and Lee (2004) measure the occurrences of similar sentences associated with the same date, whereas Pasquali et al. (2019) measure the occurrences of articles in atomic time intervals to later aggregate them and determine the bursty time periods. These statistical methods are especially suited for homogeneous datasets, but may not work as well on heterogeneous or fragmentary datasets. One could also train a Learning-to-Rank model on summaries created by experts in order to detect important sentences as in (Tran et al., 2013). This would, however, require training data which tend to be scarce, even when for contemporary data.

Alternatively, one could use an Event Detection model to detect and annotate events in the dataset as in Chasin (2010). Event Detection is another task in the NLP community that has been extensively studied, and some previous works such as Nguyen et al. (2020) have already applied these methods in humanities contexts. However, we need to keep in mind that training such a model requires annotated resources that are often lacking, especially for historical data, and that the OCR quality of documents impacts the output of these models.

Finally, we could select as event any sentence containing at least a time expression, either explicit or implicit as in Duan et al. (2019); Nguyen et al. (2014). This selection could be made even finer by taking sentences that also contain a Named Entity as in Abujabal and Berberich (2015); Bedi et al. (2017). One can then apply algorithms such as Affinity Propagation (Frey and Dueck, 2007) or Chinese Whispers (Biemann, 2006) to gather sentences describing the same event as in Rusu et al. (2014); Yu et al. (2021); Steen and Markert (2019).

Regardless of the method used to detect them, events should all be associated with time. These could be the time expressions occurring with the event mentions, or the Document Creation Date (DCD) if no time expressions are present. Alternatively, approaches for estimating the focus time of text (Jatowt et al., 2015), in absence of any temporal expressions can be applied to associate event-related sentences with particular points of time.

3.4.2 Event Importance Estimation

As mentioned in Section 2, the importance of an event can be measured in a supervised or semi-supervised manner with a classifier (Chasin, 2010; Ge et al., 2015). This method, however, requires

[2] https://impresso-project.ch/app/
[3] https://www.newseye.eu/

training data that are difficult to obtain or produce. Furthermore, the process leading a classifier to a prediction is generally not explained. Since the goal of this framework is to assist in the study of longitudinal datasets, it is necessary that the process of generating a timeline is interpretable. Thus, we would suggest to measure the importance score in an unsupervised manner by extracting features from the dataset as in (Nguyen et al., 2014; Chieu and Lee, 2004; Campos et al., 2018). Some of the features that we think could help measure this importance score are listed below, with suggestions on how to compute them:

Redundancy: The more frequently an event is mentioned, the more important it should be. One can then simply count the occurrences of events, or as an alternative, assign them importance weights by calculating their TF-IDF scores over all the time units. However, as the data might be fragmentary in archive datasets, this feature should rather not be used alone;

Contemporary references: an event may be important at a given time if other events occurring around the same period of time refer to it. Thus, to evaluate this feature, we could count how often an event is referred to from the descriptions of other events in a given short period of time around that event;

Retrospective references: Similarly, an event is likely to be important if documents keep mentioning it some time after it occurred. To assess this kind of across-time reference to the event, one could count how often (and perhaps for how long) an event is mentioned by other events that occurred after a given period of time. Other solutions may rely on computing random walks over graphs composed of timestamped events and/or entities to measure the amount of signal propagation from the past towards "the recent times" (Jatowt et al., 2016);

Causality: an event is likely to be important if it is the cause of other events that occurred after it. To evaluate the causality of an event, one could use *date reference graphs* as in Tran et al. (2015b), which measure the frequency of references, the topical influence and temporal influence between two events to determine a causal link. It is also possible to use Causal Relation Extraction (CRE) methods as presented by Gao et al. for example. However, the CRE task is far from solved and may require much more dataset pre-processing;

Common sense: some events are clearly more important than other, e.g. the birth of a child or marrying a partner are usually more important events in a family history than repainting a house. To represent that kind of common sense knowledge and compute this feature, it may be necessary to create a dataset of events that are deemed important to train a 1-class classifier (1CC) as in Duan et al. (2019) or a Learning-to-Rank model as in Ge et al. (2015). Note that while important events can be collected from historical textbooks or history-related content, gathering unimportant events may be less easy and more problematic; hence the solution could be to rely on a 1CC task.

Using these features, a straightforward formula to calculate the importance of an event could be:

$$\alpha \cdot F1 + \beta \cdot F2 + \gamma \cdot F3 + \delta \cdot F4 + \epsilon \cdot F5$$

where $F1, F2, F3, F4, F5$ are the scaled values of the features described above and $\alpha, \beta, \gamma, \delta, \epsilon$ are hyper-parameters of which value is defined by the user or document archive custodians. Similarly to event detection, the user could be asked to select any of these features to compute this score.

Some periods may contain much more documents than others. For instance, fewer documents may be available during a war time because of censorship or paper restriction. This lack of documents may lead to events that are far more or far less mentioned than others, and bias frequency-based features such as *redundancy*, *contemporary* and *retrospective references*. Thus, these features should be normalized before being incorporated.

Furthermore, we suggest these features since they are easy to compute, but we also acknowledge that they may not be sufficient to measure the importance of an event from the perspective of an expert such as a historian. Because the formula to compute the importance score is modular, one could incorporate more features in collaboration with experts.

3.5 Timeline Presentation

In this section, we describe the second main step of the framework, which generates the timeline from events scored in the previous step. We present sets of filters to select which events should appear on the timeline and how they should be presented. We also describe an optional step of timeline augmentation using external data.

3.5.1 Event Filtering

A dataset may contain hundreds or thousands of mentioned events. It is necessary to select those that will be added to the timeline. To do so, we can use filters such as described below. The weight of these filters could be changed on the user interface, thus allowing users to instantly update the timeline.

Top N: top N most important events are retained;

Importance Threshold (IT): only events of which the importance score is superior to a pre-fixed threshold IT are taken. Individual thresholds for the features described in Section 3.4 that make up the importance score can also be set;

Topical Diversity Threshold ($TopDT$): removes redundant event mentions and ensures the timeline is topically diverse. Topical diversity can be simply measured using Maximal Marginal Relevance (MMR) (Goldstein-Stewart and Carbonell, 1998) or the *n*-gram blocking metric as in Liu (2019);

Temporal Diversity Threshold ($TempDT$) : ensures every time unit on the generated timeline is evenly represented by setting a minimum and maximum number of events that can appear at each time unit.

3.5.2 Event Description Selection

There are multiple ways to represent an event on a timeline. One could select a sentence that describes the event. If this sentence is too long, one could use sentence compression methods (Filippova and Strube, 2008) to only keep its most important part. As mentioned earlier, an event might be represented by a cluster of sentences. The user can thus select one sentence among this cluster or generate a cloud of terms of all sentences contained in it, as in Duan et al. (2019). One could also use headlines if the target documents are articles as in Tran et al. (2015a); Pasquali et al. (2019).

Finally, we could also use a Natural Language Generation (NLG) system as in Steen and Markert (2019), as these generated texts are often easier to understand than text extracted from the documents. However, abstractive methods such as these may suffer from inaccuracies or hallucinations, i.e. generate information that is not present in the original documents. Thus, abstractive methods might generate improper event descriptions and lose the connection with the original documents. On the other hand, a common drawback of purely extractive methods is that selected sentences may require some context or at least post-processing for users to be able to properly understand them (e.g. pronouns may need to be resolved or we need to add definitions or descriptions of some entities or events).

3.5.3 Timeline Augmentation

To properly understand them, some events may require contextual knowledge that is missing from the processed dataset. This can especially happen if the user is not a domain expert. Such contextual knowledge may be found in knowledge bases such as Wikidata or Wikipedia Year pages (see for example (Tran et al., 2015c)). Thus, timelines generated by an ATLS system could be augmented with contextual data provided by external knowledge bases as in (Ceroni et al., 2014). These augmented timelines could help in explaining a dataset by summarizing it and providing the user with the necessary knowledge to understand it. Unfortunately, most resources created by experts are not in a machine-readable format (Gutehrlé et al., 2021). Hence, this step may require more effort.

3.6 Timeline Evaluation

As mentioned earlier, the evaluation of a TLS system is a difficult task because of the lack of evaluation datasets and the inherent subjectivity of the task. In order to evaluate the output, we would suggest to manually assess the produced timelines, either by following some evaluation criteria as in Duan et al. (2017), or by comparing them with resources created by experts such as timelines derived from history books as in Bedi et al. (2017). One could also use this framework to bootstrap an evaluation dataset specific to the given corpus, towards an automatic evaluation.

4 Discussion

In this section, we describe two hypothetical use cases comparing the application of TLS and ATLS systems, and compare in Table 1 the types of datasets and timelines both methods can process. Finally, we discuss potential extensions of ATLS.

4.1 Use cases

In the first hypothetical use case, a user has curated a homogeneous dataset of timestamped documents from Web archives. This dataset is made of news articles related to a story spanning over a year. It has been pre-processed to remove HTML tags and

	TLS	ATLS
Covered period	Shorter	Longer
Input Data Size	Small / Medium	Large
Documents type	Timestamped documents (e.g. news articles)	
Document Format	Usually born digital	Often digitized
Data Integrity	Usually complete	Can be fragmentary
Presence of noise	Less likely	Depends on OCR quality
Semantic evolution	Less common	Possible (esp. over long time)
Need for query-based filtering	Optional (depends on data size and heterogeneity)	
Need for contextualization	Less likely	More likely (esp. over long time)
Need for interpretable output	Yes	

Table 1: Comparison of TLS and ATLS tasks

extract temporal expressions. To generate the timeline, the user applies the TLS method: important dates are first selected before generating a summary of events occurring at each date. The user can select the sentence mentioning the event, the headline of the article or apply abstractive methods to generate its description.

In the second hypothetical use case, a user has curated a heterogeneous corpus to study the economical life of a certain French region in the 20th century. This corpus is composed of periodicals, newspapers and magazines from different sources (parishes, libraries, etc.) published over a century and processed with OCR. This dataset has also been pre-processed: the documents have been first cleaned of OCR errors, then automatically annotated with Temporal Expression Extraction and Named Entity Recognition components. Furthermore, the dataset has been indexed so as to allow query-based searching. To generate the timeline, the user applies some of the ATLS approaches mentioned in this paper: events are first detected by clustering similar sentences that contain a temporal expression and a Named Entity. The importance of these events is then scored using the formula described in Section 3.4.2. The timeline is generated by setting high values to the topical and temporal diversity thresholds, and augmented with external data from Wikidata, so as to ensure a comprehensive and contextualized timeline. Similarly, the user can select from the user interface to use a cloud of terms or a sentence from the cluster to generate event descriptions.

4.2 Extensions of ATLS systems

Timelines are usually represented linearly, where each time unit is of the same size (usually a day or a year). However, the optimal granularity of temporal units might vary when generating a timeline over a long period of time. For example, when referring to a distant past, humans tend to often describe entire decades or years rather than discussing each day or month which is more common for the recent past. Furthermore, events mentioned in historical documents might not always be recorded with the same temporal precision (e.g., some events may have missing dates, the dates can be imprecise or difficult to be inferred). A possible solution would be to generate logarithmic timelines, where the granularity of the time unit changes over time, as suggested in Jatowt and Au Yeung (2011).

If the documents in the datasets are annotated with Named Entities, one could generate entity-based timelines. This could help understand the history of a specific entity such as a person or a location as in Duan et al. (2019). This idea could be extended by generating aggregate timelines for multiple entities at the same time. These timelines could be agglomerative or contrastive and respectively show the similarities and differences between the history of multiple entities of the same type (e.g., cities in the same region or country, scientists of the same area). Similar to Duan et al. (2020), such comparative timelines would allow to study the history of entities of the same or similar type, e.g. Berlin vs. Paris or even entities of different types, e.g. Paris and the writer Victor Hugo.

5 Conclusion

TimeLine Summarization can be a useful tool for getting an overview of historical collections as well as it can serve as a novel information access means to news article archives. In this position paper, we have presented an overview of existing TLS methods and described a conceptual framework for Archive TimeLine Summarization systems.

The implementation of the framework outlined in this paper will be the subject of our future work. We also intend to ask humanities scholars (historians, archivists, ...) to evaluate the quality of generated timelines and the effectiveness of our framework for the study of archive collections.

References

Abdalghani Abujabal and Klaus Berberich. 2015. Important events in the past, present, and future. pages 1315–1320.

James Allan, Ron Papka, and Victor Lavrenko. 1998. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 37–45, New York, NY, USA. Association for Computing Machinery.

Omar Alonso, Stefano Marchesin, Marc Najork, and Gianmaria Silvello, editors. 2021. *Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems, Padova, Italy, September 15-18, 2021*, volume 2950 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Harsimran Bedi, Sangameshwar Patil, Swapnil Hingmire, and Girish Palshikar. 2017. Event timeline generation from history textbooks. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 69–77, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chris Biemann. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, New York City. Association for Computational Linguistics.

Emanuela Boros, Nhu Khoa Nguyen, Gaël Lejeune, and Antoine Doucet. 2022. Assessing the impact of OCR noise on multilingual event detection over digitised documents. *International Journal on Digital Libraries*.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *Advances in Information Retrieval*, pages 806–810, Cham. Springer International Publishing.

Andrea Ceroni, Nam Khanh Tran, Nattiya Kanhabua, and Claudia Niederée. 2014. Bridging temporal context gaps using time-aware re-contextualization. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 1127–1130, New York, NY, USA. Association for Computing Machinery.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Rachel Chasin. 2010. Event and temporal information extraction towards timelines of wikipedia articles.

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*, SIGIR '04, page 425–432, New York, NY, USA. Association for Computing Machinery.

Yijun Duan, Adam Jatowt, and Katsumi Tanaka. 2017. Discovering typical histories of entities by multi-timeline summarization. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, HT '17, page 105–114, New York, NY, USA. Association for Computing Machinery.

Yijun Duan, Adam Jatowt, and Katsumi Tanaka. 2019. History-driven entity categorization. In *Web and Big Data*, pages 349–364, Cham. Springer International Publishing.

Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. 2020. Comparative timeline summarization via dynamic affinity-preserving random walk. In *ECAI*.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *INLG*.

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. Modeling document-level causal structures for event causal relation identification. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Tao Ge, Wenzhe Pei, Heng Ji, Sujian Li, Baobao Chang, and Zhifang Sui. 2015. Bring you to the past: Automatic generation of topically relevant event chronicles. 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)*, pages 575–585, Beijing, China. Association for Computational Linguistics.

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*, pages 1322–1334, Online. Association for Computational Linguistics.

Jade Goldstein-Stewart and Jaime G. Carbonell. 1998. Summarization: (1) using MMR for Diversity-Based Reranking and (2) Evaluating Summaries. In *TIPSTER*.

Nicolas Gutehrlé, Oleg Harlamov, Farimah Karimi, Haoyu Wei, Axel Jean-Caurant, and Lidia Pivovarova. 2021. SpaceWars: A Web Interface for Exploring the Spatio-temporal Dimensions of WWI Newspaper Reporting. *CEUR Workshop Proceedings*.

Adam Jatowt and Ching-man Au Yeung. 2011. Extracting collective expectations about the future from large text collections. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 1259–1264, New York, NY, USA. Association for Computing Machinery.

Adam Jatowt, Daisuke Kawai, and Katsumi Tanaka. 2016. Predicting importance of historical persons using wikipedia. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 1909–1912. ACM.

Adam Jatowt, Ching-man Au Yeung, and Katsumi Tanaka. 2015. Generic method for detecting focus time of documents. *Inf. Process. Manag.*, 51(6):851–868.

Rémy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. 2012. Finding salient dates for building thematic timelines. In *ACL*.

Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. *Summarize Dates First: A Paradigm Shift in Timeline Summarization*, page 418–427. Association for Computing Machinery, New York, NY, USA.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Elvys Linhares Pontes, Ahmed Hamdi, Nicolas Sidère, and Antoine Doucet. 2019. Impact of OCR Quality on Named Entity Linking. In *International Conference on Asia-Pacific Digital Libraries 2019*, Kuala Lumpur, Malaysia.

Yang Liu. 2019. Fine-tune BERT for extractive summarization.

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*, pages 230–240, Brussels, Belgium. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Rubén Urizar. 2015. SemEval-2015 task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado. Association for Computational Linguistics.

Stephen Mutuvi, Antoine Doucet, Moses Odeo, and Adam Jatowt. 2018. Evaluating the Impact of OCR Errors on Topic Modeling. In *Maturity and Innovation in Digital Libraries. 20th International Conference on Asia-Pacific Digital Libraries, ICADL 2018, Hamilton, New Zealand, November 19-22, 2018, Proceedings*, pages 3 – 14.

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*, pages 1208–1217, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Nhu Khoa Nguyen, Emanuela Boros, Gaël Lejeune, and Antoine Doucet. 2020. Impact analysis of document digitization on event extraction. In *Proceedings of the 4th Workshop on Natural Language for Artificial Intelligence (NL4AI 2020) co-located with the 19th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2020), Anywhere, November 25th-27th, 2020*, volume 2735 of *CEUR Workshop Proceedings*, pages 17–28. CEUR-WS.org.

Thi Tuyet Hai Nguyen, Adam Jatowt, Mickaël Coustaty, and Antoine Doucet. 2021. Survey of Post-OCR processing approaches. *ACM Comput. Surv.*, 54(6):124:1–124:37.

Arian Pasquali, Ricardo Campos, Alexandre Ribeiro, Brenda Salenave Santana, Alípio Mário Jorge, and Adam Jatowt. 2021. Tls-covid19: A new annotated corpus for timeline summarization. In *ECIR*.

Arian Pasquali, Vítor Mangaravite, Ricardo Campos, Alípio Mário Jorge, and Adam Jatowt. 2019. Interactive system for automatically generating temporal narratives. In *Advances in Information Retrieval*, pages 251–255, Cham. Springer International Publishing.

Delia Rusu, James Hodson, and Anthony Kimball. 2014. Unsupervised techniques for extracting and clustering complex events in news. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 26–34, Baltimore, Maryland, USA. Association for Computational Linguistics.

Roser Saurí, Jessica Moszkowicz, Bob Knippen, Rob Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2.1.

David A. Smith and Gregory Crane. 2001. Disambiguating geographic names in a historical digital library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '01, page 127–136, Berlin, Heidelberg. Springer-Verlag.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31, Hong Kong, China. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

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*, SIGIR '00, page 49–56, New York, NY, USA. Association for Computing Machinery.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015a. Timeline summarization from relevant headlines. In *Advances in Information Retrieval*, pages 245–256, Cham. Springer International Publishing.

Giang Tran, Eelco Herder, and Katja Markert. 2015b. 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)*, pages 1598–1607, Beijing, China. Association for Computational Linguistics.

Giang Binh Tran, Tuan Tran, Nam Khanh Tran, Mohammad Alrifai, and Nattiya Kanhabua. 2013. Leveraging learning to rank in an optimization framework for timeline summarization.

Nam Khanh Tran, Andrea Ceroni, Nattiya Kanhabua, and Claudia Niederée. 2015c. Back to the past: Supporting interpretations of forgotten stories by time-aware re-contextualization. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 339–348. ACM.

Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. Multi-TimeLine summarization (MTLS): Improving timeline summarization by generating multiple summaries. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 377–387, Online. Association for Computational Linguistics.