TiCCo: Time-Centric Content Exploration

Philip Hausner∗
hausner@informatik.uni-heidelberg.de
Institute of Computer Science,
Heidelberg University
Heidelberg, Germany

Dennis Aumiller∗
aumiller@informatik.uni-heidelberg.de
Institute of Computer Science,
Heidelberg University
Heidelberg, Germany

Michael Gertz
gertz@informatik.uni-heidelberg.de
Institute of Computer Science,
Heidelberg University
Heidelberg, Germany

ABSTRACT
Time is a natural way to order information and can be utilized to summarize events and to construct a chronology of contents within a document collection in many application domains. Structuring the sequence of events along a timeline allows users to grasp information at-a-glance, which enables them to get familiar with a topic in only a short amount of time and can hence support the analysis of more complicated and heterogeneous textual data. The manual construction of timelines, however, is a tedious and error-prone task, leading to static timeline representations that limit users to a passive role. In this paper, TiCCo, an automated extraction pipeline from arbitrary English and German text collections, is provided and presented to the user in an interactive manner. This puts the user in an active role in which she not only absorbs knowledge, but also influences in which ways the information is presented to her. In-depth investigations of a specific point in time are augmented by utilizing time-centric co-occurrence graphs that further summarize information extracted from a document collection, and enable users to explore the chronology of events by allowing them to interact with the constructed graphs as well as the underlying documents.

CCS CONCEPTS
• Information systems → Document representation.

KEYWORDS
text analytics, temporal information, text exploration, automatic timeline generation, co-occurrence graph

ACM Reference Format:

1 INTRODUCTION
As the amount of documents and textual data in general is steadily increasing, average users as well as professionals are faced with an abundance of available data. However, it is usually unfeasible for users to read an entire document collection to grasp all necessary information, and even for professionals it is often beneficial to get an initial overview of a new topic before contemplating the details. Moreover, concise data representation can accelerate the work process in general, e.g., by enabling users to quickly confirm or reject certain assumptions they have about the data. With that in mind, representing textual data regarding temporal information contained in the text is often a useful approach to organize information. This is true for various domains: In the legal domain, it can help to build case chronologies; in the news domain, it can summarize important political events; and in the historical domain, it can give an overview of the events during a certain period of time. A typical way to visualize such chronological data is to employ timeline representations that give a concise overview of a time period as well as it is a well-known and intuitive tool many users are already familiar with. In this paper, we demonstrate TiCCo, a novel system to organize and explore document collections in a time-centric fashion utilizing a previously introduced graph-based approach. Thereby, we provide the following contributions:

• An extendable pipeline to automatically annotate and extract time-centric co-occurrence graphs from English or German document collections.
• An interactive user frontend that allows for an intuitive exploration of large document collections in a time-centric manner. The system includes references to the original text excerpts as well as query capabilities. Figure 1 gives a first impression of the interface.
• An open-source implementation of the proposed pipeline available on GitHub1, as well as a publicly accessible demo2 and video demonstration3.

Another important key aspect of this work is the exploratory component of the proposed system. The goal is to not only provide the user with information, but also to encourage her to actively interact with the components of a timeline. Therefore, constructed graphs contain links to the underlying documents from which the graph was built. This enables users to understand the origin of nodes and edges in the graph and gives them opportunities for further explorations as well as a deeper understanding of the applied method. Further exploratory possibilities include the querying for terms, or the access to related dates given a time-centric graph. After a summary of related work in Section 2, Section 3 details the underlying model, and Section 4 introduces the proposed implementation based on documents from the English Wikipedia related to the American Civil War.

1https://github.com/PhilipEHausner/TiCCo
2https://ticco.ifi.uni-heidelberg.de
3https://www.youtube.com/watch?v=ajiF3Kxanak
Figure 1: Overview of the user frontend showing the main interface components. The underlying data set is extracted from Wikipedia articles covering the American Civil War. On top, the user can query for terms and adjust visibility of timeline granularities. The main timeline is depicted in the center. On the bottom left, the graph associated with 7th April 1862 is shown. In the bottom right corner, co-occurrences of nodes of interested can be investigated further.

2 RELATED WORK

Because time is one of the key dimensions of human life, the processing of temporal information contained in textual data has become a widely researched field of study. Recent surveys by Campos et al. [2] and Lim et al. [5] give an introduction to the field and outline its relevance for many challenges including topics like question answering and temporal text similarity. Moreover, timelines are generally acknowledged to be a suitable representation for temporal data, and a timeline summarization aims to give a compact overview of a topic by extracting sentences or short paragraphs from document collections that best describe points in time mentioned in the data. Recently, Steen and Markert [10] introduced an abstractive timeline summarization model that computes timelines completely unsupervised using multi-sentence-compression. However, in general this is not a suitable representation for exploratory scenarios, since there is no way for the user to interact with the result. On the other hand, research by Alonso et al. [1] has shown that timelines can be used for the exploration of search results, and recently, Piskorski et al. [6] have introduced a framework for timeline generation and visualization that allows users to explore content from online news. Spitz et al. [8] presented EVELIN, a graphical interface that shows in which ways word networks can be effectively utilized for exploratory tasks, however, their implementation does not leverage temporal information to structure the underlying data. Furthermore, Przytkova et al. [7] and Spitz et al. [9] introduced time-centric graph models similar to ours, but they did not formalize their approaches in the form of a timeline or provided ways to utilize them for exploratory scenarios.

3 THEORETICAL BACKGROUND

In this section, we briefly recap the theoretical framework by Hausner et al. [3] this work is based on. First, the underlying graph model is introduced, and second, a node ranking function is presented.

3.1 Document Model

Let $P$ be a collection of documents (or pages). Each document $p \in P$ consists of a set of sentences $s \in p$. The set of all sentences is denoted by $S = \bigcup_{p \in P} \{s \mid s \in p\}$, and each sentence $s \in S$ is treated as a bag-of-words, such that the order of words is not preserved in this model. It should be noted that two sentences that contain identical words are both included in $S$, and are not treated as the same sentence in this model. For temporal exploration of data, it is furthermore of utmost importance to identify words that carry temporal information. In this context, $D$ denotes a set of dates with two dates $d_1 \in D$ and $d_2 \in D$ being equal if they refer to the same day. To account for varying temporal granularities, $D$ is partitioned into $D = D_y \cup D_m \cup D_d$, where the indices denote years, months, and days, respectively. Utilizing this partitioning, an inclusion hierarchy can be formulated: For each day, there exists a month in which it is included, and the same relation holds between months and years.

3.2 Time-Centric Co-Occurrence Graphs

Given a set of dates $D$, a time-centric co-occurrence graph is a weighted graph $G_D = (N_D, L_D)$ with nodes $N_D$ being the terms extracted in a window of $x$ sentences around timestamp $d \in D$, and links $L_D$ representing co-occurrences between the terms in the same context window. The set of all time-centric co-occurrence graphs is denoted by $G_D = \{G_d \mid d \in D\}$. It is crucial to note that not for each occurrence of a date $d \in D$, a separate time-centric graph exists, but co-occurrences around different instances of $d$ are
aggregated in the same graph \( G_d \). Utilizing the introduced inclusion hierarchy, it can be stated that for each graph \( G_d \), representing the network for a day \( d_1 \), any node \( n \) or edge \( e \) that is part of \( G_d \), also needs to be included in graph \( G_{m_1} \), for the month \( m_1 \) containing \( d_1 \). Again, the same holds for months and years, respectively.

3.3 Sentence Functionality

To extend the exploratory possibilities of the model, the sentence functionality is defined by \( \text{sent} : L_d \to \mathbb{P}(S \times S) \) with \( \mathbb{P} \) being the power set. The sentence functionality assigns to each edge all pairs of sentences in which the two words that are connected by the edge co-occur. It should be noted that the sentences need not be distinct. This allows to not only indicate the relation between two words in a graph, but also links the edge to its origins, i.e., the mutual co-occurrences of two terms in the given document collection.

3.4 Node Weighting

Each node is assigned a weight computed by the term frequency - inverse timestamp frequency (tf-itf) weighting scheme, an adaption of the popular tf-idf scheme. It is defined by:

\[
\text{tf-itf}(n, d, D) := \text{tf}(n, d) \cdot \text{itf}(n, D),
\]

with \( \text{tf} \) being the normalized term frequency in the context window around \( d \in D \), and \( \text{itf} \) defined as

\[
\text{itf}(n, D) = \log \left( \frac{M}{1 + \left| \{ d \in D : \text{tf}(n, d) > 0 \} \} \right),
\]

with \( M \) being the number of unique timestamps in the document collection. The denominator is increased by 1 to avoid zero-division.

4 DEMONSTRATION

In this section, we demonstrate the utility of our proposed interactive user interface, especially with an application for downstream exploratory tasks. We further give an introduction to the preprocessing of data.

4.1 Preprocessing

In the preprocessing step, time-centric co-occurrence graphs are created from raw text. In a first step, temporal expressions are extracted from the document collection using the temporal tagger HeidelTime [11]. Since all preprocessing is programmed using Python, this work also provides potential users with a Python wrapper for HeidelTime along with an installation script. The text is then further processed with the help of spaCy[4], which is used for sentence splitting, named entity recognition, lemmatization, and removal of out-of-vocabulary tokens, resulting in a set of sentences for each document. Time-centric co-occurrences are then extracted from this document structure, which are finally employed for the construction of the time-centric co-occurrence graphs.

4.2 Interactive User Interface

As shown in Figure 1, the user interface is divided into four major parts. A simple search interface and selector for visibility of the three different granularities, i.e., years, months and days. Further, a central timeline, accompanied by a graph interface, and lastly, a

\[\text{https://github.com/PhilipEHausner/python_heideltime}\]

Figure 2: The graph visualization interface including a dropdown menu for selection of related dates.
also explore in a bottom-up fashion, quickly identifying relevant parts of the document collection, as well as the main time frame during which an event occurred. Coupling this with co-occurrence based search further extends the capabilities of this approach. Alternatively, a starting point from a specific day gives similar insights of a narrow portion of the data set, and also allows for a quick expansion of the context into the broader frame of reference.

4.4 Data Set and Online Demonstration

For demonstration purposes, we extracted a document collection from the English Wikipedia related to the American Civil War that was fought between 1861 and 1865. The data set consists of the main article about the Civil War\(^6\) as well as all English Wikipedia articles linking to this page. The articles are then split into subsections and all documents whose title are either a date, or contain expressions as list of, references, external links, and the like, are removed. This subdivision is beneficial in certain processing steps, since HeidelTime as well as the extraction of time-centric co-occurrences performs better if only smaller sections are taken into account. For graph generation, a window size of \(x = 2\) is used, which provides a reasonable trade-off between graph size and context sensibility. The final data set consists of 211,672 articles resulting in 2,221 individual graphs between 1860 and 1865 that are used for the demonstration. The data set as well as the underlying documents can be explored using the proposed system in our online demonstration (see the first footnote). In the demonstration itself, all graphs are reduced to the 25 highest-ranking nodes with regard to the tf-idf score to minimize computation costs for users. This results in a total of 60,521 (indistinct) nodes and 339,004 edges across all graphs.

REFERENCES


\(^6\)https://en.wikipedia.org/wiki/American_Civil_War, accessed on 19 June 2020