Clustering and Exploring Search Results using Timeline Constructions

Omar Alonso
Dept. of Computer Science
University of California, Davis
Davis CA 95616, U.S.A
oralonso@ucdavis.edu

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

Ricardo Baeza-Yates
Yahoo! Research
Barcelona, Spain
rbaeza@acm.org

ABSTRACT

Time is an important dimension of any information space and can be very useful in information retrieval and in particular clustering and exploration of search results. Search result clustering is a feature integrated in some of today's search engines, allowing users to further explore search results. However, only little work has been done on exploiting temporal information embedded in documents for the presentation, clustering, and exploration of search results along well-defined timelines.

In this paper, we present an add-on to traditional information retrieval applications in which we exploit various temporal information associated with documents to present and cluster documents along timelines. Temporal information expressed in the form of, e.g., date and time tokens or temporal references, appear in documents as part of the textual context or metadata. Using temporal entity extraction techniques, we show how temporal expressions are made explicit and used in the construction of multiple-granularity timelines. We discuss how hit-list based search results can be clustered according to temporal aspects, anchored in the constructed timelines, and how time-based document clusters can be used to explore search results that include temporal snippets. We also outline a prototypical implementation and evaluation that demonstrates the feasibility and functionality of our framework.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; H.5.2 [**Information Interfaces**]: User Interfaces

General Terms

Design, Experimentation

Keywords

Temporal information, hit list clustering, exploratory search

1. INTRODUCTION

As the amount of generated information increases so rapidly in the digital world, the concept of time as dimension along which information can be organized and explored becomes more and more important. Time plays a central role in any information space, and it has been studied in other areas like information extraction, question-answering, and summarization [21]. Time and time measurements can help in recreating a particular historical period or describing the context of a document or document collection, which can be helpful for relevance purposes. As an extension to existing ranking techniques, which are primarily based on popularity or reputation, time can be valuable for placing search results along a well-defined timeline to support exploration tasks at multiple time granularities.

A look at any of the current search engines shows that temporal aspects of documents are exclusively used to sort the hit list by date, which is primarily the date a Web page has been created or last modified. In some cases, this approach can be misleading, because the timestamp is provided by a Web server and may not be accurate. Other search applications provide a range date search as part of the advanced search options. Still, the search results are filtered based on the above specific types of date attributes. Thus, for search purposes, the time dimension is mainly restricted to the metadata associated with documents and does not exploit the temporal information embedded in the documents.

Hit list clustering has emerged as an alternative mechanism to present similar documents without requiring the user to go through hundreds of items (see, e.g., [32, 33]). Clustering of result documents can lead to better user interfaces and, therefore, to an improved user experience, in particular in the context of information exploration. A study on user search experience shows, users do prefer to use clustering when they are trying to get an overview or explore a topic [8].

There are several scenarios that illustrate how valuable temporal information can be for information retrieval and/or exploration tasks on a collection of documents. Assume, for example, one would like to get an idea of a particular computer science topic by retrieving relevant papers on that topic (e.g., through Google Scholar or ACM Digital Library). If one would like to know the earliest or most recent paper(s) on that topic or even the period of time when the topic was "popular", organizing relevant documents along some kind of a timeline would be very helpful. Similar scenarios can be envisioned for exploring a news repository. For example, how would one search for news about acquisitions a company has made before a particular date or in a particular time period? The timestamp of news alone is not sufficient, as recent news documents might also talk about "old" events.

Even simple queries against Web search engines show that oftentimes organizing the documents in a hit list along some timeline can be helpful. For example, a query for [soccer world cup]

CIKM'09, November 2–6, 2009, Hong Kong, China.

against search engines now returns mostly pointers to documents that cover the last event in Germany. But every soccer fan knows that this event happens every four years. Another example is [Iraq war]; here, results are primarily related to the latest events with little from the 90's war. Wouldn't it be useful if a tool on top of a traditional retrieval system is more aware of the temporal information embedded in the documents and allows the user to have the results presented and clustered according to temporal information embedded in the documents?

As documents are the main objects that any information retrieval and exploration application has to deal with, for the above scenarios, it is important to identify diverse types of temporal information associated with documents. Words or combinations of words in a document can form *temporal expressions* that refer to some point or event in time. A temporal expression can be explicit, such as "May 20, 2007" or "December 5th, 2005", implicit, such as "New Years Eve 2006" or "Labor Day 2001", or relative to a point of narration, such as "yesterday" or "in two weeks". Every text-rich document typically contains several such types of temporal expressions. Extracting temporal expressions from the document content and *anchoring a document along a well-defined timeline* is a crucial step for any time-related document exploration task.

In this paper, we present the concepts and techniques underlying a novel document exploration framework in which we utilize temporal information extracted from the documents, and arranging documents in the form of clusters along a timeline supporting multiple time granularities. The proposed approach relies on extracting diverse types of temporal information from documents and making this information explicit in the form of *temporal document profiles*. Such profiles, obtained in a document annotation process, are used to construct timelines at different levels of time granularity. These timelines build the basis for anchoring and clustering the documents, and provide a means for exploring temporal features of document collections, such as hit list based search results.

Our approach is applicable as an add-on to any traditional information retrieval application. For example, the documents in a hit list returned by a search engine can be clustered based on the temporal information extracted from the documents. The approach is applicable to any domain specific corpora or vertical search engine where the document collection is to be organized and clustered based on the temporal information embedded in the documents.

In summary, the paper makes the following contributions.

- We present a temporal document annotation model, based on well-founded concepts of time at different granularities. We also outline a process to extract diverse types of temporal information from documents and representing such information in the form of *temporal document profiles*.

- We introduce a time-based clustering algorithm called *TCluster*, which serves as an add-on to standard document retrieval techniques, and supports the clustering and exploration of documents based on their temporal profiles. In combination with temporal snippets, we show how valuable this can be for exploration of search results.

- We outline a prototype implementation that illustrates the realization of the system architecture and document processing pipeline using standard system components and tools.

- We present an evaluation of our approach using different document collections and time-related document exploration tasks to demonstrate the feasibility and utility of the proposed techniques.

- We use Amazon Mechanical Turk (AMT[1]), a crowdsourcing platform for our evaluation studies.

The remainder of the paper is organized as follows. Section 2 discusses related work on exploratory search, hit list clustering, and temporal retrieval. A user survey on timelines is presented in 3. In Section 4, we detail the concepts of time, temporal expressions, and the time annotated document model. Section 5 presents the components of the TCluster algorithm and temporal snippets. The system architecture and prototypical implementation are outlined in Section 6. Section 7 describes the evaluation guidelines and results. Section 8 summarizes the main contributions of the paper and outlines future work.

2. RELATED WORK

There is some research on using time for a different search applications but only little work has been done on exploiting temporal information associated with documents for clustering and exploring search results [5]. The time frames project is one approach to augment news articles by extracting time information [17]. Extensions to document operations like comparing the temporal similarity of two documents in the context of news articles is presented in [20] (see also [18]). Temporal mining of blogs is presented in [27]. Recently, new research has also emerged for *future retrieval* [9] where temporal information can be used for searching the future. Using tags to visualize photos taken over a period of time is a good example of how useful time can be for arranging objects [13].

There is exciting research on adding a time dimension to certain applications like news summaries [3], temporal patterns [30], retrieval [7], and temporal Web search [24]. The special issue on temporal information processing gives a clear map of current directions [22]. Also recently, Google has added the `view:timeline`[2] feature to display search results along a timeline, allowing a limited exploration of a hit list.

Another technique related to our approach is *hit list clustering*. In general, hit list clustering groups search results into categories that are derived from the actual search [36]. Instead of processing the entire document set, hit list clustering uses a small document set that fits several well-known substring searching algorithms. Current hit list clustering engines like Vivisimo rely on a separate search engine that provides some information like Web page title, URL, and document snippets for the construction of the clusters. Rarely a temporal expression appears as part of the cluster labels. Using the Vivisimo search engine as baseline, there has been a number of approaches to improve hit list clustering, such as named entities for labeling clusters [32] or grouping by specific entities and citations [2]. A popular hit list clustering construction technique is based on suffix trees; alternatives to this technique are described by Ferragina and Gulli [14]. The recent survey by Carpineto el al. [10], covers all the latest research on Web clustering. A survey on data clustering is given by Jain et al. [16].

Very recently, crowdsourcing has emerged as a viable alternative to conduct large scale evaluation of different types of experiments for a wide range of applications like relevance evaluation [6] and user studies [19]. This recent research has shown that AMT results are reliable and very useful for gathering extra feedback, as we will discuss in our evaluations.

Research activities in exploratory search systems have gained a lot of attention lately as they add a significant user interface component that helps users search, navigate and discover new facts and

[1] `http://www.mturk.com/`
[2] `http://www.google.com/experimental/`

relationships [34, 35]. An example of placing search results in a timeline for desktop search is presented in [28]. Research on *temporal annotations* is very recent, and it is well covered in the book by Mani et al. [21]. Identification of time depends heavily on the language and the corpora, so traditional information extraction systems tend to fall short in terms of temporal extraction. Based on the latest advances, new research is emerging for automatic assignment of document event-time periods and automatic tagging of news messages using entity extraction [29].

We base our temporal document analysis, clustering, and exploration techniques on the latter approach. In particular, in contrast to most of the approaches mentioned above, we establish a solid foundation to combine the aspects of (1) extracting various types of temporal information from documents, (2) clustering and organizing document based on temporal data, and (3) visualizing such information in an exploratory search interface that helps users to study, explore and compare search results and individual documents in a temporal context.

3. EXPLORATION SCENARIOS

We start our research by conducting a series of user surveys about timelines. In the first user study, we performed a survey among 30 persons (graduate students and faculty) regarding temporal information to discover alternative exploration scenarios that can be used in future developments. The same user survey was conducted using AMT, using a crowdsourcing paradigm [6]. In total, 50 people responded to the survey.

The survey consisted of a short description of timelines and the following questions:

1. Do you think current timelines for organizing or clustering search results (such as in Google's timeline) are useful for some of your daily search activities?

2. Do you use (or would use) timelines to explore search results?

3. Please indicate some search scenarios where you use timelines or would like to use timelines to organize search results.

4. Please give some examples of search scenarios where current search engines do not sufficiently support the concept of timelines to organize and explore search results?

5. What other features would you like to see in the context of timelines?

Of all the respondents, 76% answered "yes" for question 1 and 71% answered "yes" for question 2.

For question 3, most of the search scenarios proposed by users can be classified into three main categories, given the information need as presented in table 1. Not surprising, the same categories are used to identify the lack of support in current search engines, as presented in table 2.

For the last question, we can identify presentation and exploration as the main categories where users see the value in using timelines for search. Table 3 summarizes the user feedback.

In summary, users are interested in using timelines for different information seeking tasks. They have identified limitations of current search engines to satisfy such search scenarios. In the rest of the paper, we introduce our temporal information retrieval framework that addresses some of those issues, with an emphasis on exploration of search results using timelines.

Category	User comment
History	Biographical info historical data (lists of kings, battles, etc.) - I also like time lines for past events such as happenings in World War II in order. - To know the history/achievements of a famous person - Biographies, bibliographies, referenced text publication dates, filmographies - When looking for information dealing with a specific time frame this would be useful - Rather then information on Germany itself you could search for info on Germany during WWII - History of person or country evolution of animals.
Research	When I am searching for reference articles about an object and need the oldest information first and the newest last. - Introduction of various technical protocols or software releases, or update releases, etc. -
Events	Any time where there is a list of winner of any contest or award - I like timelines to look through sporting event winners such as Super-Bowl, Daytona 500 Winners, etc.

Table 1: Summary of user feedback for question 3.

Category	User comment
History	Scientific development histories, i.e., development of the light bulb - There is no decent timelines for happenings of WII or any war. - Say for instance if we are asking for the achievements of a great person year on year some search engines provides paragraph wise data which is so difficult to interpret rather than a tabular column. I believe here the concept of timeline could be better presented.
Research	I frequently explore and research many variety of vintage items and it takes forever to find out the items history as far as the manufacturer and when that one closed and who took over, a nice timeline for this would be wonderful - Search on swine flue and get a timeliness of infected cases, plotted on a time line instead of news articles - I recently searched for volcanic activity in the US and US presidents. Both would have been better served by timelines - I like to search for info on stocks. A timeline of news about the stock might be helpful.
Events	When you ask for the winners of an award per year and the screen becomes a jumble of names.

Table 2: Summary of user feedback for question 4.

Category	User comment
Presentation	Shading and color - pop up details with mouse hover - snippets - I would like to see Bibliographies and referenced text lists in a timeline context. Regarding the internet, clicking on these items in the timeline context and linking to the referenced material would be helpful. - maybe graphs with clickable embedded links - Charts, graphs ... In other words more pictorial representation rather than textual data
Exploration	The ability to narrow fields down by date of information being presented or date uploaded would be nice. - A moving timeline or timeline that scrolls and shows short clips of events - Maybe if the dates and events have links to sites with more information.

Table 3: Summary of user feedback for question 5.

4. TIME ANNOTATED DOCUMENT MODEL

As motivated in the introduction, there is a lot of temporal information in any corpus of documents, be it ranked documents in a hit list or a corpus of topic specific documents. To take advantage of such time related information for information retrieval and in particular exploration purposes, in a document processing step, it is important to extract this information, anchor it in time, and make it explicit to subsequent document clustering and exploration tasks. In the following, we introduce the *time annotated document model* as basis for such tasks. We first describe the concept of time underlying our approach, and then discuss in Section 4.2 how temporal expressions are represented in documents and how they are anchored in time. In Section 4.3, we then introduce the concept of temporal document profiles, which make the temporal information extracted from documents explicit for time-based document clustering and exploration techniques.

4.1 Time and Timelines

As the basis for anchoring documents in time, we assume a discrete representation of time based on the Gregorian Calendar, with a single day being an atomic time interval called *chronon*. Our *base timeline*, denoted T_d, is an interval of consecutive day chronons. For example, the sequence "March 12, 2002; March 13, 2002; March 14, 2002" is a contiguous subsequence of chronons in T_d. Contiguous sequences of chronons can be grouped into larger units called *granules*, such as weeks, months, years, or decades. A grouping based on a granule provides us with a more coarse-grained timeline, such as T_w based on weeks or T_y based on years. An example of a week chronon in T_w is "3rd week of 2005".

In the following, we assume the four timelines $\mathcal{T} = \{T_d, T_w, T_m, T_y\}$ for days, weeks, months, and years, respectively. The composition of granules naturally induces a lattice structure in $\mathcal{T}$. That is, we have the relationship $T_j \gg T_i$ if timeline T_j is (transitively) composed of granules of timeline T_i. In particular, we have $T_y \gg T_d$, $T_y \gg T_m$, $T_m \gg T_d$, and $T_w \gg T_d$, but not $T_m \gg T_w$ as months are composed of days and not weeks.

Associated with each timeline $T \in \mathcal{T}$ is a *precedence relationship* $\prec_T$ that allows to compare chronons. For two chronons $t_i, t_j \in T, t_i \neq t_j$, we then have either $t_i \prec_T t_j$ or $t_j \prec_T t_i$. For example, for the two day chronons $t_i =$ "March 12, 2004" and $t_j =$ "January 5, 2004", $t_j \prec_{T_d} t_i$ holds. This precedence relationship can easily be generalized to chronons from different timelines, e.g., "March 31, 2001" is after "April, 1999".

Note that a base timeline can be defined on any other atomic time interval, such as an hour (of a day) or a week. The particular choice depends on the characteristics of the document collection to be time annotated and information exploration tasks and do not have any particular impact on the generality of the proposed approach.

4.2 Temporal Expressions

We now focus on the representation and extraction of temporal information from documents. As indicated in the introduction, a key aspect of our approach is to take a set of documents $\mathcal{D}$, extract all types of temporal information associated with documents in $\mathcal{D}$, and make this information available to our document clustering and exploration techniques.

The first type of such information is the *document metadata*, which appears as the date a document $d \in \mathcal{D}$ has been created or last modified. We denote such time related metadata of a document $d \in \mathcal{D}$ as *document timestamp* $d.ts$. Document timestamps typically can be obtained at document collection (crawling) or indexing time and can be anchored in the timeline T_d.

The second type of temporal information is a little bit more involved as it relates to the linguistic analysis of the textual content of documents. A suitable approach to identify time in text data is named-entity extraction, with *temporal entities* being time-related concepts. Such concepts are represented in the document text as (not necessarily contiguous) sequences of tokens or words. In particular, temporal entities can be made explicit in the form of a *temporal expression* that correspond to a chronon in some timeline. The expressions are recognized by an entity extraction approach using a time-based linguistic analysis. In the following, we concentrate on the more conceptual aspects of the types of temporal entities and their representation as temporal expressions. A realization of this approach in a prototype using some advanced techniques is presented in Section 6.

Contrary to other entities such as names and places, temporal entities can be represented as temporal expressions that are sequences of not necessarily contiguous tokens.Expressions can be mapped to temporal entities and terms defined in some temporal ontology. Similar to the approach by Schilder and Habel [29], we distinguish between explicit, implicit, and relative temporal expressions.

Explicit temporal expressions describe chronons in some timeline, such as an exact date or year. For example, "December 2004" is an explicit expression that is anchored in the timeline T_m. Similarly, the expression "September 12, 2005" is anchored in T_d. Depending on the capabilities of the entity extraction approach and in particular its underlying time ontology, *implicit temporal expression*, such as names of holidays or events can be anchored in a timeline as well. For example, the token sequence "Columbus Day 2006" in the text of a document can be mapped to the expression "October 12, 2006", which is anchored in T_d. Is is also conceivable that a single temporal entity can be mapped to more than just one temporal expression, i.e., several chronons at once. For example, the token sequence "Winter season 2005" can be mapped to a combination of month, week, and day chronons in three different timelines. In general, implicit temporal expressions require that at least a year chronon appears in the context of a named event.

Relative temporal expressions represent temporal entities that can only be anchored in a timeline in reference to another explicit or implicit, already anchored temporal expression. That is, their anchoring depends on a chosen point of time reference or narration. For example, the expression "today" alone cannot be anchored in any timeline. However, it can be anchored if the document is known to have a creation date as a reference. Then it is likely that the expression can be mapped to that date. There are many instances

of implicit temporal expressions, such as the names of weekdays (e.g., "on Thursday") or months (e.g., "in July") or references to such points in time like "next week" or "last Friday". Relative temporal expressions may even include more vague temporal information. Instances of these include phrases such as "in a few weeks" or "some years ago". In general, there is less confidence in determining relative temporal expressions than in explicit or implicit expressions, an aspect that becomes important when all temporal expressions discovered in a document are represented in a temporal document profile, as illustrated in the next section.

Although it might seem almost infeasible to detect and in particular anchor relative temporal expressions, there have recently been significant advances in detecting and mapping instances of various types of relative temporal information. Again, we will discuss respective techniques and tools in Section 6. Here we are more concerned with the representation of the different types of anchored temporal expressions in documents, as detailed next.

4.3 Temporal Document Profiles

In the following, we describe how explicit, implicit, and relative temporal expressions determined by a named-entity extraction approach are made explicit for our timeline construction and document clustering approach. In our time annotated document model, this process of entity extraction is a function denoted tdp, for *t*emporal *d*ocument *p*rofile. It associates with each document d of a document collection $\mathcal{D}$ a list of 3-tuples. It has the signature

$$tdp\colon D \rightarrow [E \times C \times P]^*$$

where the set $E = E_e \cup E_i \cup E_r$ denotes the set of explicit, implicit, and relative temporal expressions E_e, E_i, and E_r respectively. C denotes the set of chronons from timelines in $\mathcal{T} = \{T_d, T_w, T_m, T_y\}$. The set P denotes the set of positions of temporal expressions in a document. A position is a composite of the number of the sentence in which the expression occurs, the position in that sentence, and the absolute position of the expression in the document. More precisely, for a particular mapping

$$\begin{aligned} d \rightarrow [&(e_1, c_1, p_1), \ldots, (e_{k_e}, c_{k_e}, p_{k_e}),\\ &(e_{(k_e+1)}, c_{(k_e+1)}, p_{(k_e+1)}), \ldots, (e_{k_i}, c_{k_i}, p_{k_i}),\\ &(e_{(k_i+1)}, c_{(k_i+1)}, p_{(k_i+1)}),\\ &(e_{(k_i+2)}, c_{(k_i+2)}, p_{(k_i+2)}), \ldots, (e_{k_r}, c_{k_r}, p_{k_r})] \end{aligned}$$

where *tdp* is applied to a document $d \in \mathcal{D}$, resulting in a list of 3-tuples, the meaning of the components of the triples is as follows:

1. The first k_e tuples describe the explicit temporal expressions that have been determined in d, together with their normalized chronons and positions in d (we will elaborate on normalization below).

2. The next $k_e + 1$ to k_i tuples describe implicit temporal expressions, again with their normalized chronons and positions in d.

3. The $k_i + 1$ tuple corresponds to the timestamp $d.ts$ of the document d. It is assumed that every document has such a timestamp. Depending on the confidence one has in the document timestamping approach, the timestamp can be considered either an implicit or explicit temporal expression. For example, if it is known that the document creation times are exact, then the document timestamp should be considered as an explicit temporal expression.

4. Finally, the remaining $k_i + 2$ to k_r tuples describe relative expressions, again together with their normalized chronons and positions.

In the following, for a document $d \in D$, we denote the list of 3-tuples of the *temporal document profile* of d as $tdp(d)$.

There are some important properties of a temporal document profile that need to be recognized. First, all chronons $c_i, i = 1 \ldots l$, are *normalized*. That is, all chronons that are elements of the same timeline $T \in \mathcal{T}$ have the same format. For example, all day chronons that have been associated with temporal expressions are represented in the day/month/year format, such as "15/04/1966". Similar formats are assumed for months, years etc. This normalization step has some problems of its own like normalizing time zones, which we do not address in this paper as they are not essential to our approach. As we will see in Section 5, respective normalized chronons will serve as labels for document clusters when documents are assigned to clusters along a timeline.

Second, a chronon $c \in T$ for some $T \in \mathcal{T}$ can be associated with many explicit, implicit, and relative temporal expressions. In fact, the same chronon can even occur several times in a single profile $tdp(d)$ but then at different positions in the document d.

In summary, a temporal document profile makes explicit all temporal expressions in a document, specified as chronons of well-defined timelines in $\mathcal{T}$. These chronons are then used to construct a timeline for the documents at different levels of time granularity and to cluster documents along these timelines.

5. TIMELINE CONSTRUCTION AND DOCUMENT EXPLORATION

In the following, we detail how a given collection of documents with temporal document profiles is organized along a multiple-granularity timeline in the form of document clusters. This approach can be applied to either an unranked collection of documents or documents in a hit list. However, in the following, we will focus on documents in a hit list to further exploit the ranking of the documents in the hit list.

The time-based clustering algorithm, called *TCluster*, described below is realized as add-on to any standard document retrieval technique. It can be invoked automatically by the retrieval technique or by the user, if she desires a time-based exploration of retrieval results. We assume that for a query term q against a document collection $\mathcal{D}$, the retrieval algorithm determines a hit list $L_q = [d_1, d_2, \ldots, d_k]$ of k documents. We also assume that all documents in $\mathcal{D}$ have a unique id and that there is an inverted index file that allows to return a set of documents for a given query term or temporal expression. Given such a hit list, the temporal document profiles are used to construct a *time outline* for the documents first. The documents are then clustered along this timeline, again based on their document profiles. A single cluster can be thought of as a bin that contains only documents with temporal expressions matching the cluster label, which corresponds to some chronon. The organization of clusters along a timeline as well as the lattice structure imposed among timelines (cf. Section 4.1) then allows for the exploration of document clusters and a hit list, respectively, at different levels of time granularity. The following four sections elaborate on the individual steps of the *TCluster* algorithm.

5.1 Constructing a Time Outline

The first step in organizing documents along a multiple-granularity timeline is to construct a *time outline* for the documents in the hit list L_q. For this, all chronons are extracted from the temporal document profiles of the documents in L_q. We denote this multi-set of chronons $ch(L_q)$, defined as follows:

$$ch(L_q) := \{\{c \mid d \in L_q \text{ and } (e, c, p) \in tdp(d)\}\}$$

Note that the elements in $ch(L_q)$ may come from different timelines. More importantly, among the elements in $ch(L_q)$ there are (not necessarily unique) minimum and maximum chronons, which describe the *lower and upper bound* of the time outline for the documents in L_q. Based on the range between lower and upper bound, a time granularity for the time outline is chosen. Assume, for example, L_q contains a document with a temporal expression mapped to the year 1974 (as lower bound) and another document with a temporal expression mapped to the year 2007 (as upper bound). Then the temporal range of L_q is several years, and T_y is chosen as time outline for L_q. On the other hand, if L_q contains documents, say some news articles, whose temporal range is only a few weeks, then T_w would be more appropriate as time outline. In general, a time outline is a timeline representation that describes the *temporal range of documents* in L_q, independent of the "temporal distribution" of documents along this timeline. We will elaborate more on the distribution aspect below. Without loss of generality, in the following, we assume a time outline based on the timeline T_y.

5.2 Document Clustering

In the next step of the *TCluster* algorithm, the timeline chosen as time outline for L_q is used to *normalize* the chronons in $ch(L_q)$, here according to T_y. That is, if a chronon c in $ch(L_q)$ is of a granule finer than year, only the year component of c is used. In general, we denote such a type of normalization of a chronon c based on a time granule $g \in \{y, m, w, d\}$ as $norm_g(c)$. For example, $norm_y(\text{“}15/4/1966\text{“}) = \text{“}1966\text{“}$, and $norm_m(\text{“}15/4/1996\text{“}) = \text{“}4/1996\text{“}$). The *labels* for the initial (most coarse-grained) document clusters for L_q and time granule g are then determined by the following set.

$$ch_g(L_q) := \{norm_g(c) \mid d \in L_q \text{ and } (e, c, p) \in tdp(d)\}$$

Intuitively, for the timeline T_y, $ch_y(L_q)$ simply contains a set of years such that there is at least one document $d \in L_q$ that has this year (as part of) its chronons. Assume there are l cluster labels $y_1, y_2, \ldots, y_l$, $y_j \in T_y$ in $ch_y(L_q)$ among which the precedence relationship $\succ$ holds. The documents in a cluster y_j, denoted $cluster(y_j)$, are then determined as follows:

$$cluster(y_j) := \{d \mid d \in L_q, \exists c : (e, c, p) \in tdp(d) \text{ such that } norm_y(c) = y_j\}$$

Obviously, if a document $d \in L_q$ has several components in its temporal document profile $tdp(d)$, it can be in several document clusters. This is intuitive, as in the text of d, there can be many explicit, implicit, and relative references to different points in time. The question then, however, is whether there is a *main cluster* for each document in L_q. Such a cluster can easily be determined based on the distribution of the chronons in $tdp(d)$ with respect to the temporal range of the hit list L_q. For example, if the chronons associated with d refer to n different years, the main cluster for d, denoted $c\_main(d)$, would be the year for which d has the most chronons.

Finally, there can be a wide variation among the number of documents in each cluster. Some clusters might only have a very few documents whereas others have many documents, perhaps representing some kind of *hot spots*. For some years in T_y there might also be no documents at all that refer to these years. In general, from a user interface and document exploration point of view, if the time outline for L_q is represented to the user, each cluster is not only labeled by a year chronon but also shows information about the number of documents in that cluster. Referring back to the example of a topic search in a corpus of technical documents mentioned in the introduction, the cluster(s) with the most documents would then present the years in which the topic was "hot". In Section 6, we will elaborate on how properties of such document clusters as well as main clusters for documents are represented in a respective user interface.

5.3 Ranking Documents in a Cluster

Thus far, the *TCluster* algorithm only determined the sets of documents belonging to each cluster along a timeline, here T_y. Clearly, documents in a cluster should be ranked to reflect the relevance of documents in $cluster(y_j)$ with respect to *both* the cluster label y_j *and* query terms q. While the ranking of documents in a hit list L_q is solely based on q, the ranking of documents in $cluster(y_j)$ now also takes the cluster label y_j and thus time into account. Key to such a ranking is the *distance* of the query terms q to the temporal expressions in the documents in $cluster(y_j)$.

Assume a document $d \in cluster(y_j)$ with temporal document profile $tdp(d)$. Let $match_e(d, y_j)$ denote the number of times the query term q occurs *together* with an explicit temporal expression e, $(e, c, p) \in tdp(d)$, *in a sentence* in d such that $norm_y(c) = y_j$. Analogously, let denote $match_i(d, y_j)$ and $match_r(d, y_j)$ denote the number of matches with respect to implicit and relative temporal expressions in $tdp(d)$, respectively.

It is clear from the description of the $match$ functions that the more often q occurs together with explicit temporal expressions matching y_j in d's sentences, the more relevant document d is in that cluster. Furthermore, there are several reasonable choices for a ranking that combines $match_e$, $match_i$, and $match_r$. For the *TCluster* algorithm, we choose the following function:

$$rank(d, y_j) := match_e(d, y_j) + \delta_i * match_i(d, y_j) + \delta_r * match_r(d, y_j), \quad \delta_i, \delta_r \in [0, 1]$$

That is, the sentence level co-occurrence of q with an implicit or relative temporal expression matching y_i can be weighted less than a respective co-occurrence with an explicit temporal expression. In order to deal with scenarios in which no single temporal expression in d matches y_j, we simply assume that $rank(d, y_j) = \delta_i$, because every document has at least a document timestamp as temporal expression (although this does not occur with q).

The above ranking function obviously leaves much room for further investigations. For example, the $match$ functions could be extended to not only look at sentences in which the query term q occurs but also close-by sentences. Also, as implicit temporal expressions typically contain at least a year expression (plus the name of some event), even without a time ontology to match the exact day or period of that event, the year can be determined with high confidence; therefore, δ_i is typically close to 1. The choice of the δ_r heavily depends on the capabilities of the temporal entity extraction approach, in particular its ability to determine and correctly anchor relative temporal expressions.

Based on the above discussion, we use the following ranking approach in our *TCluster* algorithm. Given two document $d, d' \in cluster(y_j)$. d is ranked higher than d' in $cluster(y_j)$, denoted $d >_{y_j} d'$, if either of the following two conditions holds:

1. $rank(d, y_j) > rank(d', y_j)$
2. $rank(d, y_j) = rank(d', y_j)$ and d is ranked higher in L_q than d'.

It should be noted that due to such a "time-based" ranking in $cluster(y_j)$, a document d can be ranked higher than a document

d' in $cluster(y_j)$ although d' is ranked higher than d in the original hit list L_q. There are several obvious scenarios that make this point more clear. For example, if in document d' the query q occurs more frequently than in d, then d' is ranked higher in L_q than d. But if only in d the query q occurs together with the chronon y_j, then there is a closer relationship between q and y_j in d than in d'. This "re-ranking" is an important property of the *TCluster* algorithms and reflects the algorithm's focus on temporal information extracted from documents and represented in temporal document profiles.

5.4 Cluster Exploration

The construction of a time outline and the time-based clustering of hit list documents can be invoked by the user after a standard document retrieval and ranking approach in an exploratory search interface (see Section 6.2). For example, it can be applied to the hit list provided by a search engine. The document clusters organized along a timeline representing the temporal range of the documents and the ranked documents within each cluster now serve as the basis for a user-driven, time-based cluster exploration.

As discussed in Section 4.1, the composition of time granules leads to a lattice structure among timelines. This lattice structure is employed when the user wants to explore a given document cluster. The exploration typically starts with the timeline constructed in the first step of the *TCluster* Algorithm (cf. Section 5.1), here clusters with year chronon labels $cluster(y_j)$, and occurs in a "drill-down" fashion.

Assume a cluster $cluster(y_j)$ with ranked documents $[d_1, d_2, \ldots, d_l] \subseteq L_q$. The cluster can be *refined* based on a timeline T with $T_y \gg T$, i.e., T can be T_m or T_d. That is, the chronon is expanded to its constituent granules, providing the user with a drill-down operation. Assume the user chooses the timeline T_m to refine a particular $cluster(y_j)$. The *TCluster* algorithm then first constructs a (now more fine-grained) time outline for this particular cluster and its documents, based on months. Then, the documents in $cluster(y_j)$ are partitioned into clusters labeled by month/year chronons, and ranked in respective clusters. These two tasks follow the same procedures as described in Sections 5.1 and 5.2. The only difference now is that in the clustering step, the chronons associated with documents in $cluster(y_j)$ are normalized by the granule month and not year anymore, thus leading to a different set of cluster labels, namely months in the given year y_j. The ranking of the documents in the resulting month-based clusters m_k ($1 \leq k \leq 12$) is done in the same fashion as described in Section 5.3. Choosing finer and finer timelines clearly leads to an effective way to explore and compare document clusters and the ranking of documents in clusters, all functions that can effectively supported by a simple user interface, as described later.

5.5 Temporal Snippets

Having available the information about temporal expressions in documents, our aim now is to utilize this information and provide users with relevance cues about documents in a hit list and clusters, respectively, and subsequent exploratory search tasks. For this, we introduce the concept of temporal snippets, which are aimed at leveraging this temporal information.

Similar to traditional snippets, a temporal snippet can be considered as a document preview but one that outlines the main events in a document, which are described using temporal expressions. Such time-centered document preview can be used in many search and exploration contexts, such as exploring a hit list or just browsing a collection of temporally rich documents. For the construction of temporal snippets, the most relevant sentences from a document that contain temporal expressions need to be determined and suitably "assembled". The utility of this feature is that we want to present snippets that show why a document is in a cluster. This means that the snippet (in the ideal case) should contain the query term and the cluster label.

The TSnippet algorithm consists of candidate sentence selection and sentence ranking for producing adequate snippets. A description of temporal snippets and a user evaluation is presented in [4].

6. PROTOTYPE

We now present the realization of our timeline construction and time-based document clustering approach. In our prototype, we use a combination of existing technologies to demonstrate the feasibility of our approach. The prototype realizes two main components: (1) a back-end that supports corpora processing and storage as well as index creation, and (2) a processing unit that realizes query processing and timeline construction and also implements the user interface to the system.

Corpora processing involves taking a document collection (or hit list from a search application) and processing the documents using the Alembic [1] part of speech tagger (POS tagger). Next, the GUTime [15] temporal tagger recognizes the extents and normalized values of temporal expressions, producing an XML document. We use an Oracle10*g* database for storing the documents that have been annotated using XML. That is, temporal document profiles are embedded within the original documents as XML markup and are not managed outside the documents.

For query processing, the system takes some user query terms as input, executes the query, i.e., it performs some document ranking, and runs the *TCluster* algorithm on the search result hit list. The search results are then presented along a time outline in a Web-based user interface. The presented timeline is composed of clusters (labeled by year chronons) and provides the user with means for a more fine-grained cluster exploration. In the next section, we provide more details about the actual document annotation process realized in our prototype.

6.1 Document Annotation Pipeline

We explore the identification of temporal expressions in more detail by providing a document-processing pipeline that includes a number of operations as follows. Given a set of documents, the first step is to extract time-related metadata from the documents. This is either the creation or last modified date for a document file or the timestamp provided by the Web server for Web pages. The second step is to run a POS tagger on every document. The tagger determines the parts of speech assigned to each word/token like noun, verb etc. in a document. The tagger also tags sentence delimiters, which are needed for temporal document annotation. The third step is to run a temporal expression tagger on the POS-tagged version of the document. This step makes the temporal expressions, which are based on the TimeML standard [25], in a document explicit by producing an XML-like document. That is, resulting documents contain temporal document profile information as XML markup.

In a parallel step, a traditional entity extraction component extracts named-entities that later augment the document index. An index is created for the documents to allow to efficiently search for text as well as temporal expressions and other named entities.

6.2 Exploratory User Interface

The user interface to our prototypical system is realized as a Web-based user interface (see Figure 1). In the interface, a user

can enter query terms and explore the resulting document clusters returned for a query.

In the prototype, the year timeline is used as initial time outline for the result documents and clusters, respectively. Using the $match$ functions, the implementation of *TCluster* also determines a ranking of the documents in each cluster (cf. Section 5.3). For this, the sentence level distance between the query and temporal expressions is used.

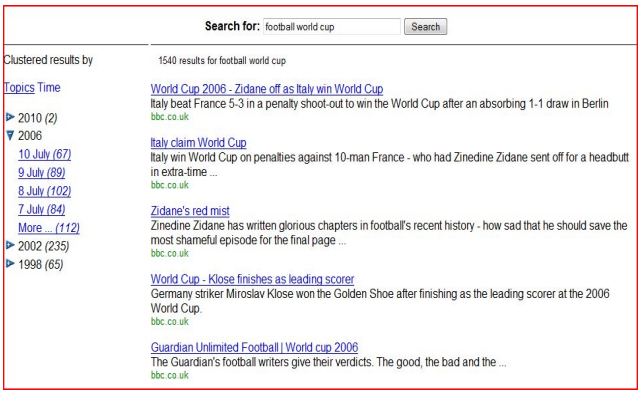

Figure 1: Timeline cluster for the query [football world cup]

Information about the document cluster is sent back to the user interface client and displayed along a timeline in descending year order. As can be seen in Figure 1, for each cluster (labeled with a year chronon), the number of documents in the cluster is shown. A cluster can be expanded, allowing the user to explore the documents in a cluster based on more fine-grained timelines, according to the clustering exploration step described in Section 5.4.

Compared to similar exploratory search interfaces like the one presented by Ringel et al. [28], our system differs in a number of ways. Instead of using an "object" timestamp (e.g., a document, email message, or presentation), we leverage both temporal expressions and document metadata, because document timestamps alone can often be misleading. We also do not restrict the search space to a personal desktop environment, because we believe that timelines should be an integral part of information exploration applications as add-ons to standard retrieval and search applications.

For exploratory search systems, taking advantage of temporal expressions embedded within a document leads to a much richer framework for exploration. We believe this is an important ingredient for the *information forager* who is trying to see the profit in terms of the interaction cost required to gain useful information for an information source [23]. Users tend to prefer sources (in this case search engines) that are richer in good results. These good results involve adding important nuggets, such as time information.

We also constructed another interface that leverages Twitter search results. Instead of listing all twitts sorted by recency, we can cluster twitts according to the timestamp. In this particular case, there is no need to use any of the processing pipeline due to the short text content (only 140 characters) so we use the timestamps of every post to cluster the hit list. Still, we can map the stream of twitts to a more meaningful user interface, as shown in Figure 2.

7. EVALUATION AND RESULTS

The primary objective of our evaluation here is to show the clear advantage our temporal clustering approach has over existing document clustering approaches that only perform document clustering based on document timestamps. We also aim to show that clustering results determined by our approach are indeed more meaningful than such standard clusters and also allow for flexible document and cluster exploration based on different timelines and time granules, respectively.

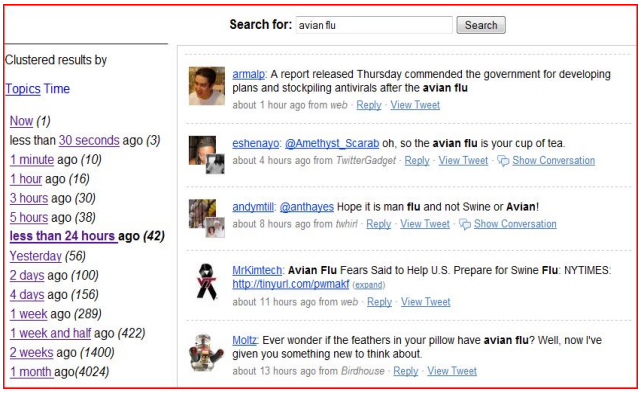

Figure 2: Timeline cluster for [avian flu] twitts.

7.1 Evaluation Guidelines

There has only been little work published on evaluation for automatically generated timelines. We adopt some of the evaluation guidelines for automated testing that are outlined on Cronopath [11]. There are two important features of a timeline that we are interested in: *precision* and *presentation*.

Precision is defined as the fraction of the retrieved documents that are relevant. The main point here is that all relevant documents must be included in the timeline. The timeline should also contain appropriate labels for each time granule. Presentation involves displaying the timeline in a graphical form. The most intuitive way is to draw a line and mark the main time granules from left to right in ascending order, thus forming a time outline.

We conducted a number of experiments with our time-based document clustering approach and prototype. The first set of experiments uses part of the well-known Internet directory DMOZ [12]. The second set of experiments uses a document collection in which temporal expression have been manually annotated using TimeML. The last set of experiments include relevance assessment for queries using a graded relevance scale.

7.2 DMOZ

DMOZ is a multilingual open content directory, which is constructed and maintained by a community of volunteers. For the experiment, we took the DMOZ category "World Cup" in the larger category soccer/football. The "World Cup" category contains a collection of articles that have been put into context by users, meaning that each document corresponds to either of the World Cups in 2010, 2006, 2002, 1998 and 1994, leading to a user-specified collection of five document clusters. We used a crawler to obtain all documents in this category and applied the document processing pipeline presented in Section 6.1 and *TCluster* algorithm to compare the number of clusters obtained through *TCluster* with the 5 pre-defined clusters in this category. In terms of precision of the timeline, the number of pre-defined categories in the World Cup category (5) is lower than the number of clusters determined by *TCluster* (21)

While this DMOZ entry has only 5 categories that represent the past four tournaments and the forthcoming one, *TCluster* determined 16 more clusters. Why this discrepancy? It is clear that each World Cup document has a single event as the main theme.

Thus, documents are well classified by users in terms of the actual event. On the other hand, the same document usually refers to past events and winners. Therefore, it is also possible to view the same document content along the time line of the history of the World Cup, meaning that a single document can be relevant to more than just one single event and thus cluster and year.

7.3 TimeBank

The second set of experiments uses the TimeBank 1.2 corpus, which contains news articles that have been annotated using TimeML with temporal expressions related to events, times and temporal links between events and times [26]. The objective of the experiments was to show that temporal expressions, if correctly identified and made readily accessible to *TCluster*, can significantly improve the precision and granularity of timelines along which documents are clustered. For comparison purposes, we run a set of search queries against this annotated TimeBank document collection and the corresponding collection that had no such annotations and marked-up temporal expressions, respectively. For the latter "plain text document collection", only the document timestamps have been used for the *TCluster* algorithm.

Intuitively, for the plain text collection, documents in the search result are clustered based on their document timestamp. Applying *TCluster* to the plain text documents should at least result in the same number of clusters, as *TCluster* considers the document timestamp as part of a temporal document profile. However, if there are more temporal expressions accessible to *TCluster* in a temporal document profile, the resulting timeline along which documents are clustered should be more precise by providing: a) more clusters in a timeline, and b) more documents in a cluster, as a document then can belong to more than one cluster.

We selected 20 random queries and compared the results. Overall, the usage of temporal expressions shows a 50% increase in the number of clusters discovered by *TCluster*. The more temporal expressions are explicit (besides just the document timestamp), the better the precision of the document clustering.

7.4 Relevance Evaluation using AMT

We conducted a second experiment on AMT where the goal was to evaluate the quality of search results using *TCluster* in combination with temporal snippets. In this experiment, we selected 10 random informational queries for Wikipedia featured articles[3]. Each query was evaluated by 11 different workers and presented as a task in AMT that consisted of search results and hit list clustering for the query. We asked workers to evaluate if the search results were relevant using the following scale (excellent = 5, I don't know = 1):

- Excellent. I can explore the results by time and I can see the timeline of events.
- Good. The search results are very relevant but there might be better results.
- Fair. Somewhat relevant. There are some items that are inaccurate.
- Not relevant. The search result is not good because it does not contain any relevant information.
- I don't know. I can't evaluate the quality of the search results.

[3]http://en.wikipedia.org/wiki/Featured_Article

The average response was 4.04% (with an 80% agreement level), which indicates that workers found results to be good most of the time. We also performed the same evaluation but using the top-10 most active topics on Twitter where the average response was 4.33% (with an 80% agreement level).

7.5 Additional User feedback

A very useful feature of the experiments in AMT is the ability to ask for feedback. For both data sets (Wikipedia and Twitter), users provided useful comments. Table 4 gives a summary of some of the answers.

Experiment type	User comment
Wikipedia	I do a lot of research and time lines of cases and sports events would be extremely useful. Being able to simply click on the relevant date would make the research move along a lot quicker.
Twitter	The results are great. I have looked for such search facility but didn't find one in Twitter. - I like the idea. What about a more visual timeline/graph though? Like, the avian flu example would have a "spike" a couple of weeks ago.

Table 4: Summary of user feedback for AMT experiments

The results are encouraging and user feedback has been very valuable in terms of alternative scenarios and overall suggestions for improvements.

8. CONCLUSIONS AND FUTURE WORK

Temporal information embedded in documents in the form of temporal expressions provide an import means to further improve current search engines and thus user experience by simply applying an additional step to traditional document ranking approaches.

In this paper, we have presented a framework that can be used to make a search application time-aware by providing more features that leverage time. We have shown how such a time-based document clustering can be achieved when temporal expressions in documents are readily available. Our novel approach leverages recent developments in temporal entity extraction and temporal document annotations, techniques that are readily available to support our approach.

Our *TCluster* algorithm provides great flexibility and allows users to explore clusters of search result documents that are organized along well-defined timelines, supporting different levels of time granularity. The prototypical implementation shows that the proposed technique can effectively be realized using existing tools and systems. Our evaluation demonstrate the utility of the time-based clustering over existing approaches that cluster documents only based on document timestamps.

Furthermore, our approach significantly leverages the work done in the context of temporal entity extraction and the development of respective tools. Our well-founded timeline construction and document clustering approaches put these tools and techniques to an important new use. In particular, our framework can serve as an offspring of several refined, topic specific methods for temporal information retrieval.

An important aspect of our work is that we believe that when a user is engaged in tasks that require time-related investigations and sensemaking, traditional information retrieval and search engines

fall short. The use of time and timelines for clustering and browsing nicely fits exploratory search systems that go beyond simply returning some documents or answer in response to a query.

Our future work primarily concerns investigating alternative document ranking techniques in *TCluster* using more (annotated) document collections. In particular, using machine learning techniques, we want to study the weighting of relative temporal expressions as well as different sentence distance functions for determining the rank of documents in a cluster.

AMT is a viable tool for conducting relevance evaluations experiments using crowds. As shown in previous research, it is possible to recruit hundreds of users at very small cost within a short timeframe. We plan to continue working on using this type of methodology with more interactive experiments to continue studying timebase search and exploration.

9. REFERENCES

[1] Alembic: `http://www.mitre.org/tech/alembic-workbench/`

[2] R. Al-Kamha and D. Embley: Grouping Search-Engine Returned Citations for Person-Name Queries. *In 6th ACM International Workshop on Web Information and Data Management (WIDM 2004)*, ACM, 96–103, 2004.

[3] J. Allan, R. Gupta and V. Khandelwal: Temporal Summaries of News Topics. In *Proc. of the 24th International ACM SIGIR Conference*, ACM, 10–18, 2001.

[4] O. Alonso, R. Baeza-Yates, and M. Gertz: Effectiveness of Temporal Snippets. *WSSP Workshop*, WWW Madrid, 2009.

[5] O. Alonso, M. Gertz, and R. Baeza-Yates: On the Value of Temporal Information in Temporal Information Retrieval. *SIGIR Forum*, 41(2):35–41, 2007

[6] O. Alonso, D. E. Rose, and B. Stewart: Crowdsourcing for Relevance Evaluation *SIGIR Forum* (42):2, 12–18, 2008.

[7] I. Arikan, S. Bedathur, and K. Berberich: Time Will Tell: Leveraging Temporal Expressions in IR. *WSDM Late Breaking Results*, Barcelona, 2009.

[8] A. Aula, N. Jhaveri, and M. Kaki: Information Search Re-access Strategies of Experienced Web Users. In *Proc. of the 14th World Wide Web Conference*, ACM, 583–592, 2005.

[9] R. Baeza-Yates: Searching the Future. In *SIGIR Workshop MF/IR*, 2005.

[10] C. Carpineto, S. Osinski, G. Romano, and D. Weiss: A Survey of Web Clustering Engines. In *ACM Computing Surveys*, 41(3), 2009.

[11] R. Catizone, A. Dalli, and Y. Wilks: Evaluating Automatically Generated Timelines from the Web. In *5th International Conference on Language Resources and Evaluation*, 2006.

[12] DMOZ `http://www.dmoz.org/`.

[13] M. Dubinko et al.: Visualizing Tags over Time. In *Proc. of 15th World Wide Web Conference*, ACM, 193–202, 2006.

[14] P. Ferragina and A. Gulli: A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. In *14th International Conference on World Wide Web (Special interest tracks and posters)*, 801–810, 2005.

[15] GUTime, `http://complingone.georgetown.edu/~linguist/`

[16] A. Jain, M. Murthy, and P. Flynn: Data Clustering: A Survey. *ACM Computing Surveys*, 31(3):264-323, 1999.

[17] D. Koen and W. Bender: Time Frames: Temporal Augmentation of the News. *IBM System Journal*, 39(4):597–616, 2000.

[18] P.J. Kalczynski and A. Chou: Temporal Document Retrieval Model for Business News Archives. *Information Processing & Management* 41, 635–650, 2005.

[19] A. Kittur, E. H. Chi, and B. Suh: Crowdsourcing User Studies with Mechanical Turk. In *Proc. 26th SIGCHI Conference on Human Factors in Computing Systems*, 453–456, 2008.

[20] J. Makkonen and H. Ahonen-Myka: Utilizing Temporal Expressions in Topic Detection and Tracking. In *Research and Advanced Technology for Digital Libraries*, LNCS 2769, Springer, 393–404, 2003.

[21] I. Mani, J. Pustejovsky, and R. Gaizauskas (Eds.): The Language of Time. Oxford University Press, 2005.

[22] I. Mani, J. Pustejovsky, and B. Sundheim: Introduction to the Special Issue on Temporal Information Processing. *ACM Trans. on Asian Language Inf. Processing*, 3(1):1–10, 2004.

[23] P. Pirolli: Information Foraging Theory. Oxford University Press, 2007.

[24] M. Pasca: Towards Temporal Web Search. *ACM Symposium on Applied Computing*, 1117–1121, 2008.

[25] J. Pustejovsky et al.: TimeML: Robust Specification of Event and Temporal Expressions in Text. New Directions in Question Answering, *AAAI Spring Symp.*, 28–34, 2003.

[26] J. Pustejovsky et al.: TimeBank 1.2 Documentation `http://timeml.org/site/timebank/documentation-1.2.html`

[27] A. Qamra, B. Tseng, and E. Chang: Mining Blog Stories Using Community-Based and Temporal Clustering. In *Proc. 15th ACM International Conference on Information and Knowledge Management*, ACM, 58–67, 2006.

[28] M. Ringel, E. Cutrell, S. Dumais, and E. Horvitz: Milestones in Time: The Value of Landmarks in Retrieving Information from Personal Stores. In *IFIP TC13 International Conference on Human-Computer Interaction*, 2003.

[29] F. Schilder and C. Habel: From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *ACL'01 Workshop on Temporal and Spatial Information Processing*, 1–8, 2001.

[30] B. Shaparenko et al.: Identifying Temporal Patterns and Key Players in Document Collections. In *Proc. IEEE ICDM Workshop on Temporal Data Mining: Algorithms, Theory and Applications* (TDM-05), 165–174, 2005.

[31] TimeML 1.2.1 Specification: `http://www.timeml.org`

[32] H. Toda and R. Kataoka: A Search Result Clustering Method using Informatively Named Entities. In *7th ACM International Workshop on Web Information and Data Management* (WIDM 2005), ACM, 81–86, 2005.

[33] Vivisimo, `http://www.vivisimo.com`.

[34] R. White, K. Kules, S. Drucker, and M. Schraefel (Eds). Supporting Exploratory Search. *Communication of the ACM* 49(4), April 2006.

[35] R. White, G. Marchionini and G. Muresan: Evaluating Exploratory Search Systems: A Special Topic Issue of Information Processing and Management. *Information Processing and Management*, 44(2), 433-436, 2008.

[36] O. Zamir and O. Etzioni: Web Document Clustering: A Feasibility Demonstration. In *Proc. of 21st International ACM SIGIR Conference*, ACM, 46–54, 1998.