Foundations and Trends® in Information Retrieval
Vol. 9, No. 2 (2015) 91–208

DOI: 10.1561/1500000043

Temporal Information Retrieval

Nattiya Kanhabua
L3S Research Center
Hanover, Germany
kanhabua@L3S.de

Roi Blanco
Yahoo Labs
London, United Kingdom
roi@yahoo-inc.com

Kjetil Nørvåg
NTNU
Trondheim, Norway
Kjetil.Norvag@idi.ntnu.no

Contents

1 Introduction **92**

1.1 Temporal Dynamics . . . 93

1.1.1 Content and Structure Changes . . . 94

1.1.2 Changes in User Behavior . . . 95

1.2 Scope and Aim of this Survey . . . 96

2 Processing Dynamic Content **98**

2.1 Adaptive Crawling . . . 99

2.1.1 Web Page Evolution . . . 100

2.1.2 Incremental Crawling Policies . . . 101

2.2 Temporal Indexing . . . 102

2.2.1 Incremental Indexes . . . 103

2.2.2 Versioned Indexing . . . 103

2.2.3 Processing Temporally Qualified Queries . . . 105

2.3 Caching Evolving Results . . . 107

2.4 Summary . . . 112

3 Temporal Information Extraction **113**

3.1 Document Creation Time . . . 114

3.1.1 Content-based . . . 115

3.1.2 Non-Content-based . . . 119

3.2 Document Focus Time 120
3.3 Entity and Event Evolution 121
3.4 Summary 123

4 Temporal Query Analysis 125
4.1 Temporal Query Intent 128
4.1.1 Mining Temporal Patterns in Query Streams 128
4.1.2 Determining Temporal Intent from Top-k Results . 136
4.2 Dynamic Query Subtopics 144
4.2.1 Mining Subtopics from Historical Query Logs . . . 145
4.2.2 Mining Subtopics from a Document Collection . . 146
4.3 Time-aware Query Enhancement 148
4.4 Summary 156

5 Time-aware Retrieval and Ranking 157
5.1 Recency-based Ranking 158
5.2 Time-dependent Ranking 163
5.3 Event and Entity-aware Ranking 168
5.4 Summary 169

6 Applications of Temporal Information Retrieval 171
6.1 Existing Temporal Search Engines 171
6.2 Analysis and Exploration over Time 175
6.3 Temporal Summarization 176
6.4 Temporal Clustering of Search Results 177
6.5 Future Event Retrieval and Prediction 178

7 Conclusions and Outlook 180

Appendices 183

A Research Resources 184

References 189

Abstract

Temporal dynamics and how they impact upon various components of information retrieval (IR) systems have received a large share of attention in the last decade. In particular, the study of relevance in information retrieval can now be framed within the so-called *temporal IR approaches*, which explain how user behavior, document content and scale vary with time, and how we can use them in our favor in order to improve retrieval effectiveness. This survey provides a comprehensive overview of temporal IR approaches, centered on the following questions: *what* are temporal dynamics, *why* do they occur, and *when* and *how* to leverage temporal information throughout the search cycle and architecture. We first explain the general and wide aspects associated to temporal dynamics by focusing on the web domain, from content and structural changes to variations of user behavior and interactions. Next, we pinpoint several research issues and the impact of such temporal characteristics on search, essentially regarding processing dynamic content, temporal query analysis and time-aware ranking. We also address particular aspects of temporal information extraction (for instance, how to timestamp documents and generate temporal profiles of text). To this end, we present existing temporal search engines and applications in related research areas, e.g., exploration, summarization, and clustering of search results, as well as future event retrieval and prediction, where the time dimension also plays an important role.

N. Kanhabua, R. Blanco, and K. Nørvåg . *Temporal Information Retrieval.*
Foundations and Trends$^{®}$ in Information Retrieval, vol. 9, no. 2, pp. 91–208, 2015.
DOI: 10.1561/1500000043.

1

Introduction

During the last decade, information retrieval has been successful in providing everybody with easy access to the vast amount of information available on the Web. As illustrated in Figure 1.1, creating, handling, and sharing information on the Web has seen the unprecedented growth and change in recent years. Cornerstones for such development are new technical devices and corresponding changes in our everyday behaviors. Digital photos and videos create large data volumes and numerous artifacts. Participative content generation and sharing in Web 2.0 solutions and social interaction via networks and platforms have gained wide acceptance, ranging from media-specific sharing (e.g., Flickr) over text and video distribution channels (e.g., Twitter and Youtube) up to web-based documentation and sharing of nearly complete life histories as encouraged by Facebook.

While the current way of accessing the Web comprise a good baseline, an optimal access to the evolving Web requires new models and algorithms for retrieval, exploration, and analytics which go far beyond what is needed to access the current state of the Web. This includes taking into account the time dimension, structured semantic information available on the Web, as well as social media and network information.

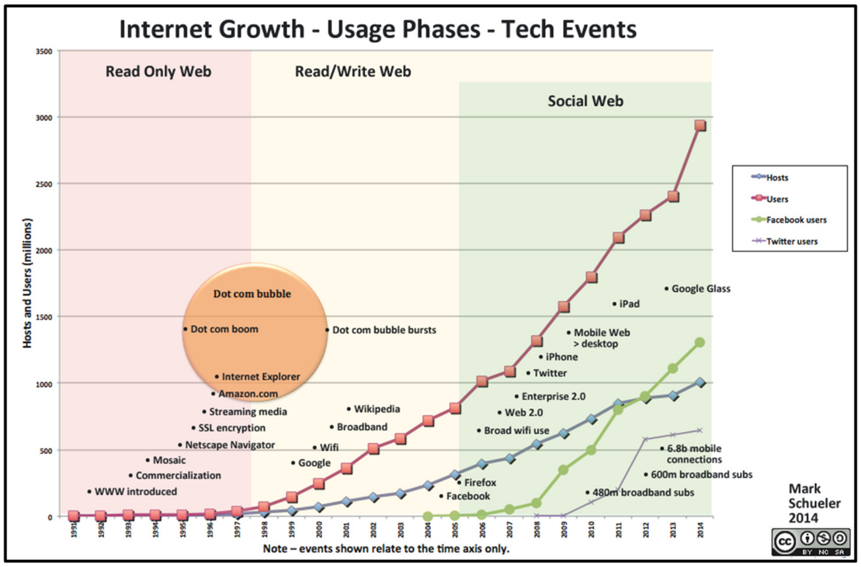


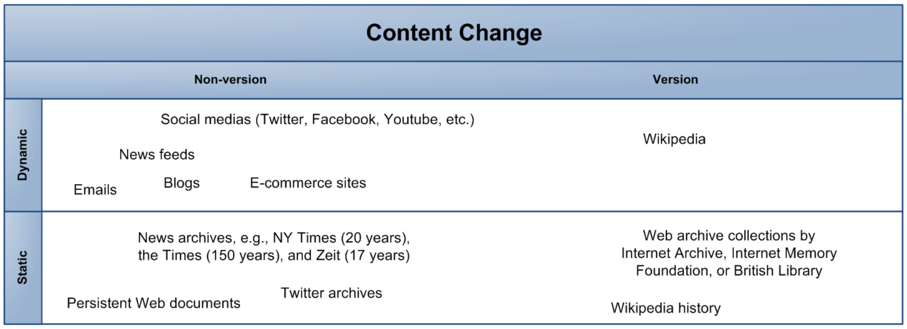
Figure 1.1: Internet Growth/Usage Phases/Tech Events (created by Mark Schueler, used with permission).

1.1 Temporal Dynamics

It is noteworthy that the time dimension has strong influence in many domains, e.g., Topic Detection and Tracking (TDT) (Allan et al., 1998; He et al., 2007), and Emerging Trend Detection (ETD) (Berry, 2003). However, in this context, we focus on the impact of time on the Web and we explain the evolution of the Web and its impact on web search and data mining before going into the details of temporal information retrieval. We will then discuss the scope and aim of this survey, and present the organization of the rest of the survey.

The Web has evolved in many aspects including its size, content, structure, and how it is accessed by people through web search engines. Such evolution has been previously discussed in (Ke et al., 2006; Risvik and Michelsen, 2002).

In this work, we aim at providing a comprehensive survey and answers to the following questions: *what* are temporal web dynamics, *why* do they occur, *when* and *how* to leverage the time dimension through-



Content Change

	Non-version	Version
Dynamic	Social medias (Twitter, Facebook, Youtube, etc.) News feeds Emails Blogs E-commerce sites	Wikipedia
Static	News archives, e.g., NY Times (20 years), the Times (150 years), and Zeit (17 years) Persistent Web documents Twitter archives	Web archive collections by Internet Archive, Internet Memory Foundation, or British Library Wikipedia history

Figure 1.2: Categorization of documents by the degrees of content change, i.e., static or dynamic.

out the search cycle and architecture. For this purpose, we begin by explaining the general and wide aspects associated to temporal web dynamics, namely, the evolution of the Web categorized by its changes of 1) content and structure, and 2) user querying behavior.

1.1.1 Content and Structure Changes

The content of the Web, changes constantly over time, e.g., web documents are added, modified or deleted continuously. National and international initiatives have recognized this need and started to collect and preserve parts of the Web (Gomes et al., 2011; Costa et al., 2013). The most prominent one is the Internet Archive, which has collected more than 456 billion web pages (as of April 15, 2015) since 1996. Two important European initiatives include 1) the Internet Memory Foundation providing a set of smaller crawls for specific topics, domains and projects and 2) the British Library that aims at preserving national web content.

As illustrated in Figure 1.2, we categorize document collections, such as, personal homepages, corporate websites, Wikipedia articles and blogs, with respect to the various degrees of change, and whether the document creators or web sites keep different versions of each hosted document, i.e., versioning vs. non-versioning. On one hand, web archives are created by periodically visiting and crawling publicly available web pages. A web archive contains documents with multiple versions since the new version of a document will be added into the archive repository when re-crawling. On the other hand, a web archive can have just one or the latest version for each document due to non-versioning policies, e.g., news archives, or real-time web data, such as, Twitter messages.

In parallel to content changing, the link structure of the Web also evolves (Dai and Davison, 2010a). The changes of content and structure affect basic processes like crawling and indexing, but also the computation of graph-based authority measures used for document ranking or spam detection.

1.1.2 Changes in User Behavior

Temporal web dynamics are related to user querying behavior in at least two ways. First, search traffic for particular queries varies over time and might present certain temporal patterns, such as, spikes, periodicity (e.g., weekly or monthly), seasonality and trends. Examples of sporadic or spiky queries are *breaking news* (e.g., iran, japan, earthquake), *celebrities* (e.g., beyonce, lady gaga), and *short-span events* (e.g., marathon, lollapalooza). Periodic or seasonal queries are, for instance, *annual events* (e.g., earth day, march madness, april fools' day, pgatour) and *television series* (e.g., american idol, crystal bowersox, dancing with the stars). Queries representing trends consist of *anticipated events* (e.g., iphone 7, mlb, miss usa), *past recent events* (e.g., easter ideas, final four), and *current events* (e.g., tax extension, presidential candidates).

Second, many queries are *time-sensitive queries*, which contain underlying temporal information needs that do not exhibit a temporal pattern in search streams. In other words, a time-sensitive query can be inferred to a particular time period, for example, an initial query Brazil FIFA World Cup might be later reformulated as 2014 FIFA World Cup. We categorize such queries with underlying temporal information needs into two types: 1) an explicit temporal query having temporal criteria explicitly provided by users (Berberich et al., 2007; Nørvåg, 2004), and 2) an implicit temporal query with no temporal criteria provided (Campos et al., 2012a; Kanhabua and Nørvåg, 2010a). An example of explicit temporal query is U.S. Presidential election 2016, whereas an implicit temporal query, e.g., Brazil FIFA World Cup, is likely to refer to the most recent World Cup event in 2014 or the historical event in 1950. Note that, the temporal intent of the latter type can be determined using temporal information extraction techniques. Several studies of real-world user query logs have shown that temporal queries comprises a significant fraction of web search queries. For example, Zhang et al. (2010) showed that 13.8% of queries contain explicit time (Nunes et al. (2008) reported 1.5%) and 17.1% of queries have a temporal intent implicitly provided (7% reported by Metzler et al. (2009)).

Understanding temporal search intent is a challenging task that is the first step for applying an appropriate time-aware ranking method. In addition to the change in information needs, user interactions in the social Web are highly dynamic over time, e.g., comments, likes, interests as well as users' profiles. This affects how user interests/profiles should be modeled by taking into account such dynamics.

1.2 Scope and Aim of this Survey

This survey gives a comprehensive overview of the most important aspects of temporal information retrieval. It describes techniques involved in the complete pipeline of processing, from obtaining web documents, document processing and indexing, information extraction, and querying. It also gives an overview of application areas showing that its use extends well beyond simply searching web archives.

In addition to giving an extensive overview, we also intend that this survey should be self-contained enough to be used as lectures/teaching material for researchers that want to get acquainted with the research area. The survey can be read and understood by anybody with basic information retrieval knowledge, but should also be of use for more advanced researchers wanting to understand in more detail this field of research. As such it extends previous overviews of challenges and opportunities in temporal information retrieval (Alonso et al., 2011b), and the survey by Campos et al. (2014b).

The remainder of this survey is organized as follows: Section 2 describes research problems for the pre-processing step of temporal document collections, i.e., dynamic crawling and temporal indexing of web documents. Section 3 presents current approaches to identifying and extracting of temporal information useful for leveraging in temporal information retrieval. Section 4 describes approaches to determining the temporal intents of queries, the effect of terminology changes over time, as well as query performance prediction for temporal queries. Section 5 describes a comparison of different time-aware ranking methods. Section 6 presents applications in information retrieval and related research areas where the time dimension also plays an important role, e.g., temporal analytics and exploration, temporal summarization, temporal clustering of search results, and future event retrieval and prediction. Section 7 concludes the survey and discusses possible research topics beyond what have been addressed in the survey. Finally, in Appendix A, we present existing research resources and recent evaluation workshops organized in the field of temporal information retrieval.

2

Processing Dynamic Content

Search engines are able to answer user queries within very limited time budgets mainly thanks to two datastructures: indexes and caches. Indexes expose the contents of a particular document collection in a term-by-term fashion, this is, they are able to return quickly which documents contain a certain set of terms. A ranking algorithm uses this information to return an ordered list of results that are deemed as relevant for a given query. Caches, on the other hand, store the results of frequently issued queries and are able to return those results with a substantially lower processing cost than accessing an index and computing those results on the fly. Prior to building the index structure, search engines need to be fed with documents (web pages in the case of a web search engine) that are acquired in an incremental polling process known as crawling.

All these three main building blocks, i.e., crawling, indexing and caching, are impacted by changes in web pages. Modern systems that keep up-to-date with web page changes need to adapt their basic processes in order to not be stale, i.e., the information contained in the engine's index and cache is not in sync with the current online web page status.

Search engines are often described as large systems operating with *static* data structures, most notably their index, which are built in *batch mode*. In practice this means that crawling, indexing a document collection and serving queries operate within generations; while the next batch of content is being generated, the search engine can only dispatch views of documents coming from the current generation. Once the next generation is ready, it replaces the old one. In practice, this would imply that the index might be stale for weeks (Cho and Garcia-Molina, 2000).

Real-life modern search engines keep (at least some) portions of their index data structures up-to-date, with a lower latency. Systems that serve specialized content that might be trending, such as Social media or News sites, strive to ingest and process new documents between minutes or less. This includes the process of re-visiting some sites to discover new or updated pages and updating dynamically the index to be able to reflect those changes as soon as possible. This process requires data structures and algorithms that can handle changes immediately rather than being replaced in batches. This section describes how search engines are able to cope with those changes, paying special attention to three main building blocks: crawling, indexing and caching.

In particular, crawlers need to keep up with page updates and tier pages automatically into different layers depending on how frequently they are updated. Similarly, retrieval systems that provide search capabilities over different versions of documents require specialized indexing structures to cope with multiple time dependent collection views. Finally, search engines make extensive use of caches in order to speed up the retrieval process. We review some alternatives to design a cache that invalidate entries given that the underlying index is changing.

2.1 Adaptive Crawling

The purpose of crawling is to gather a collection of useful web pages as quickly and efficiently as possible, while providing at least the required features for respecting the limitations imposed by publishers (*politeness*) and avoiding traps (*robustness*) (Olston and Najork, 2010). Crawlers that operate at web scale require a properly coordinated cluster of machines. There are a number of fundamental issues regarding the parallelization of the different tasks involved in the crawling process. Even if most approaches make use of a centralized manager to decide which URLs to harvest next, which is possible to perform in a fully distributed fashion (Boldi et al., 2004).

In addition, crawlers need to maintain a balance between *coverage*[1] and *freshness*[2] relative to the current web copies Olston and Najork (2010). In order to do so, they must assign their resources prioritizing which pages to crawl (or re-crawl) next. Next, we describe how to estimate that a page content has been modified and the crawler needs to re-download it to obtain a more recent snapshot. Cho and Garcia-Molina (2000) introduce two main strategies for repeated downloads:

- Batch crawling: the process is restarted periodically and within each batch there are no repeated downloads of the same page.

- Incremental crawling: pages might appear subsequently during the crawling process, which never terminates.

Most modern web search engines perform incremental crawling, although batch crawling might be more practical for systems that operate in smaller focused domains. A comprehensive review of this and other related practical aspects related to web crawling can be found in Olston and Najork (2010).

2.1.1 Web Page Evolution

A central issue for effective temporal-aware crawling relates to web page evolution from both a site-level perspective and a page-level perspective. The former refers to how many pages appear and disappear from a site and the latter to how often a particular page modifies its content. Ntoulas et al. (2004) provide insights on site-level evolution, based on the study of 150 different web sites over one year. Some key findings are that there is a rapid turnover of web pages in terms of their birth and death along with a higher turnover of hyperlinks. In particular, it seems

[1] Coverage is the fraction of desired pages that the crawler acquires successfully.

[2] Freshness is the degree to which the acquired page snapshots remain up-to-date.

that new pages are created at a rate of 8% per week whereas every week 25% new hyperlinks are created. After a year, however, around 80% of the links and web pages are replaced by new ones.

In contrast, around 70% of pages modify less than 5% of their content, measured with cosine distance using the TF-IDF of the words of the page as the vector components, after a week and 50% after one year. It seems that there is no straight correlation between the frequency with which a page changes and the amount of its content that changes over time. On the contrary, the amount of content that changes within a page has a strong correlation with the amount of content that will change in the future, although the strength of this correlation varies from site to site. Researchers have further looked into which regions of web pages are more likely to change their content over time. A large part of those changes only reflect in a small chunk of the page (Fetterly et al., 2003), and the main theme of the page tends to remain the same along time (Adar et al., 2009). Additionally, most web pages can be classified into static, churn, i.e., old content is being replaced by new one, and scroll, i.e., the new content is appended to the page, like in blog posts. Most pages contain a static part (like the boilerplate text) and therefore a large portion of the web is a mixture between different types, only varying the rate of churn and scroll.

2.1.2 Incremental Crawling Policies

In order to cope with web dynamics, web crawlers must tradeoff coverage and freshness by prioritizing fetching unseen or previously visited pages. In many cases how to balance between these two objectives is decided on a use case basis. Systems that aim to capture content changes over time have to create a *model* of the temporal behavior of web pages, this is, estimate when a page will have new content. Then, they have to allocate their crawling (bandwidth, machines) resources accordingly and produce a *schedule* or crawl order that follows the target revisits as close as possible. A good model of temporal behavior has to be able to estimate at which point in time a page will change given samples of content, history of updates and information about similar pages.

Cho and Garcia-Molina (2003b) propose different frequency estimators based on Poisson processes and investigate their bias, consistency and effectiveness. In practice, this means that changes can be tracked through a number of discrete events. For instance, the simple estimate would be the number of times we observe change in a page over the number of times we visited the page. More involved estimators take into account the date of last modification and the period of time between changes, which might not be evenly spaced in time. By using those estimators, a crawler might improve the ratio of fresh pages by as much as 35%. Other works consider estimators using the content of web pages (Barbosa et al., 2005), or frequency of updates of pages with similar content, link structures and other features (Cho and Garcia-Molina, 2003a).

We note that most of these approaches work independently of the definition of “page” and “change”. For instance, one might engineer a crawler so that it captures pages that have changed more than 30% of their content, or to revisit not web pages but rows in a database (modern distributed data stores like HBase or Google’s BigTable (Chang et al., 2006) fold versioning directly into index structure). These techniques can be applied provided that the notion of “crawled unit" is clear and that there is a mechanism to detect whether there has been a change in that unit or not.

2.2 Temporal Indexing

In the previous section, we explain that crawlers need to keep up with page updates and tier pages automatically into different layers depending on how frequently they are updated. On the other hand, whenever a web page modifies its content it has to be reflected in the search engine index as soon as possible. Similarly, retrieval systems that provide search capabilities over different versions of documents, such as web archives, require specialized indexing structures to cope with multiple time dependent collection views. We review different approaches to process efficiently keyword queries over the changing indexes.

2.2.1 Incremental Indexes

Once a retrieval system is equipped with a mechanism to continuously update its contents, we need to be able to reflect those changes whenever a user query is submitted. In turn, this requires that the underlying data structure serving the results is aware of these changes. Most modern information retrieval systems employ *inverted indexes* as their basic structures for storing and organizing documents' contents (Baeza-Yates and Ribeiro-Neto, 2011). The process of adding, removing or modifying the documents present in the index is known as dynamic or online indexing. Given that indexes are compressed and information is stored in contiguous chunks of memory, the process of index updating adds a layer of complexity into the indexing process. Index updates are generally handled by keeping a smaller in-memory index which is periodically merged with a larger index (that might reside in main memory or on disk) whenever a memory threshold is reached Lester et al. (2008).

2.2.2 Versioned Indexing

How to index versioned archives requires of even more complex indexing and maintenance procedures. The main difference with respect to online indexing is that all the previous versions of a document are maintained and ready to be queried. Versioned indexing comes from the database field (Anick and Flynn, 1992) although it has a long story within the IR community (Nørvåg and Nybø, 2005, 2006). The approaches are built upon the idea of maintaining, along with the current document, small delta updates for every version, each one of them with their own creation date timestamp.

Earlier approaches worked with corporate help-desk corpora as a first example domain with evolving data (Anick and Flynn, 1992). Their solution consists of time-stamped delta entries that are comprised of modified text, with additions or deletions. The index can be rolled-back to a particular point in time by first rolling back the status of the document at a desired timestamp. Then, the query can be evaluated on that particular time instance of the index. The number of rollback actions drives the execution time in this approach which can be very inefficient for archives that span a large number of updates. It is possible, however, to index redundant content in an efficient manner by identifying content that can be stored just once and shared among several documents. Therefore, the fact that the different versions contain a high degree of duplicate content can be exploited to produce much more compact indexes (Herscovici et al., 2007) if the overlapping text is identified appropriately.

The problem of eliminating redundancy in index structures is closely related to that of reducing space or network transmission costs in storage systems by exploiting redundancy in the data. The delta-versioned based index organization gives efficient access to more recent versions, but costly access to older versions. A popular way to make the access cost uniform regardless of the age of the replica is to rely on a post-process filtering. The approach operates in two stages, first performing a regular search ignoring the temporal component and secondly filtering search results according to any specified temporal constraints (Nørvåg and Nybø, 2005, 2006). There are a number of supplementary data structures that can be employed to speed up this filtering process. For instance, Nørvåg and Nybø (2005) propose to store a map from a document version identifier to its corresponding time period, and Nørvåg and Nybø (2006) suggest to replace the term to version mappings of postings with term to version-range mappings. In any case, once a query is submitted into the system their complete postings lists must be examined regardless of the query time span.

Berberich et al. (2007) introduce an additional schema to handle point queries over temporally versioned document collections. The approach extends the standard index posting lists with a time-frame value indicating the version of the document in which the term is present. These lists might end up being extremely large, and therefore one must resort to additional pruning and compression techniques to keep the index manageable. Berberich et al. (2007) present two different posting reduction options, *temporal coalescing* or merging sequences of postings that contain the same document id and similar (comparable) scores, and *sublist materialization*, which divides every posting list into several sublists according to time intervals. These are lossy compression schemes constrains the index structure with respect to the scoring mechanism, which must be predetermined in advance before building the index.

In the case the temporal index includes positional information, i.e., in which in the document the terms occurred, one needs to avoid repeated indexing of substrings that are common to several versions of a document. This can be done by partitioning every document version into a number of fragments, which are then indexed individually. At query processing time, every document fragment matching the query has to be joined in order to determine which version contains the query words. Broder et al. (2006) organize the different related documents into a tree structure in which each node represents a document. Each node distinguishes between content that it is unique to the document and content that it is shared across versions. The latter is inherited from each ancestor node. On the contrary, Zhang and Suel (2007) use string partitioning techniques to split documents into fragments. This has the advantage of allowing for highly efficient updates.

These techniques consider each version as a separate document, or assembly artificial virtual documents from several versions. The work by He et al. (2009) takes a slightly different approach by modeling a versioned collection using a two-level index, a document level where each distinct document has a global identifier and a version level index, which specifies which of the versions of the document contain a particular term. This allows for effective compression techniques at the version level when the set of different replicas is represented by a bitvector of length equal to the number of versions of the document.

2.2.3 Processing Temporally Qualified Queries

A related important issue is how to process efficiently keyword queries over a temporally partitioned inverted index, or over a collection of timestamped documents. The latter case is a particular instance of *filtering* or *faceting*. We can assume that documents contain a special numeric field that represents the date, although without loss of generality this field can represent sizes of a product catalog, opening times of business listings among a large array of possibilities. The queries that can be issued to such an index could be comprised of a text portion as well as constraints on the values of one or more numeric fields. Numeric constraint consists of an allowed range of values per some specific numeric field. These can be one-sided $(-\infty, 40)$, or ranged $(-10, 20)$. In the case of dates, if we assume the field contains normalized timestamps with respect to some time units, queries are of the form $t_1 \leq t \leq t_2$, where t_1 and t_2 are valid timestamps.

There is a large body of work by the database community that is focused on numeric or string attributes but that ignores large sets of documents and inverted index structures. Apart from temporally qualified queries, one could consider similar use cases in search engines that include support for numerical range constraints and techniques for geographic or local search. The index needs to know how to store numerical attributes and provide methods to filter using the numerical expression. General purpose inverted indexes maintain a list of occurrences of terms in documents, along with some additional information (positions, term frequency) and a simple extension to this schema is to add a *payload* to each posting (Fontoura et al., 2007). Another option would be to include additional external structures for accessing per-document metadata, that should be kept in main memory to avoid incurring in a large number of I/O operations per query that could make the processing time too large for practical purposes.

Additionally, processing temporally qualified queries might incur a larger query overhead than processing plain keyword queries. As we have seen, versioned indexing introduces additional data structures and requires extra filtering processes to determine which version of a document matches a particular query. In this case a large fraction of reads from the inverted lists do not contribute to the final result, namely all the entries that do not qualify for the temporal predicate of the query.

Berberich et al. (2007) address the problem of answering time qualified queries that refer to a single point in time, given that it is sufficient to consider just one temporal partition for each term at query processing time. Answering a temporally qualified query in the case of indexes that have been augmented and partitioned according to time information involves choosing the correct index partitions to evaluate the query. To speed up query processing of more general classes of temporal queries (those that have a *time range* as a query qualifier), Anand et al. (2010) propose an algorithm that selects partitions of the index that would return the highest number of matching documents. Then, partitions that contain documents with entries that are already contained in those high-recall partitions can be simply not retrieved, saving I/O operations. However, intuitively, one could make use of the temporal range specified in the query *before* matching documents against query terms, and then perform retrieval in a sub-set of documents that comply with the time constrain. For instance, a simple approach could consist of partitioning the index into a large number of time ranges and then performing the search only on the valid partitions. This has the drawback of affecting negatively the compression ratio of the index posting lists, although even this simple schema using state of the art compression techniques is able to yield reasonably efficient query response times. It is further possible to organize the index in a way that allows fast access to postings that satisfy the temporal constraint (He and Suel, 2011).

In all the previous approaches inverted indexes augmented with time information are sliced along the time-axis, this is, each partition of the index contains documents that fall within a previously specified time interval. As we mentioned, this might blow-up the index size. This can be partially alleviated using adequate compression schemes. Taking an orthogonal approach, Anand et al. (2011) propose to partition horizontally in shards each index list along the document identifiers axis instead of time. This way, there is a small overhead with respect to index size. Queries are then processed by opening all the list shards per query term and seeking the right time position within each shard and then scanning the posting list sequentially from that position. There is also a way to calculate the optimal numbers of shards given the cost of the first random access compared to the cost of sequential access.

2.3 Caching Evolving Results

Caching of search results is one crucial optimization step in modern search engines. They employ some dedicated fixed-size fast memory cache that can store a certain number of result pages. When queries are submitted, the system performs a cache look-up and if the query has been previously served (hit) it can quickly return the result pages to the user. Whenever the results are not cached (miss), the engine evaluates the query and computes its results. The results are returned to the user and stored in the cache. Whenever the cache is full, a *replacement policy* takes place and decides which queries stored in the cache might be flushed to make room for the newly computed set. The primary goal of cache replacement policies is to exploit the *locality of reference* that is present in many real life query streams in order to exhibit high hit ratios and lower the number of costly query evaluations. Roughly speaking, locality of reference means that after an object x is requested, x and a small set of related objects are likely to be requested in the near future.

Caching is a technique that fundamentally relies on the presence of historical reliable data in order to achieve its optimal results (Fagni et al., 2006) given that a large portion of popular queries are submitted by several users. Query streams are extremely redundant and bursty; query popularities follow a heavy tailed distribution, implying that there exists a high level of redundancy, and some topics are trending between some windows of time, implying that query frequencies might result in spikes or bursts over the time axis. Caching yields immediate benefits; firstly, it shortens the engine's response time (users have to wait less to see their results delivered) and secondly it lowers the number and cost of query executions (which in turn results in a lower number of machines needed to keep up with a certain user and query traffic). Most caching replacement policies are exclusively tailored for search engines that do not modify their index contents over time (Fagni et al., 2006). Furthermore, a fundamental underlying assumption of caching applications is that subsequent submissions of a query will result in the same response and returning the cached entry does not degrade the application. This assumption is broken when dealing with an incremental indexing situation given that the corpus is constantly being updated. Therefore the results of any query can potentially change at any time.

Within this context, a search engine has to face the problem of deciding whether to re-evaluate repeated queries, thereby reducing the effectiveness of caching their results, or to save computational resources at the risk of returning stale (outdated) cached entries. This results in a tradeoff between *freshness* and *number of computations performed.* On one extreme, the search engine would not cache anything at all; in this case every returned set of results would be *fresh* (or, the results would reflect the most recent state of the index), but then the engine would have to perform a great number of redundant operations. On the other extreme, the engine would never invalidate the results cache in the presence of index updates, not performing any redundant work at all (as determined by the replacement policy) but a great deal of the results would be stale and will not reflect the current index state.

Many search-based systems apply simple solutions to this dilemma, ranging from performing no caching of search results at all to applying time-to-live (TTL) policies on cached entries so as to ensure worst-case bounds on staleness of results (Cambazoglu et al., 2010). In practice, this means that each entry in the cache is timestamped and this TTL value reflects the time point after which the result entry will be considered stale. On the contrary, Blanco et al. (2010) propose a mechanism to invalidate selectively cached results from queries whose results are actually affected by the updates to the underlying index. The main intuition is that not *every* document will change continuously, so one can devise selective mechanisms to detect which cache entries are unaffected by document updates. More precisely, the idea is that when a document change is reflected in the index, every cached query that return this document should be invalidated. The task is cast as a prediction problem, in which a component (a *cache invalidator predictor*) that is aware of both the new content being indexed and the contents of the cache, invalidates cached entries it estimates that have become stale.

The main components of the indexing and caching and their interaction flow is depicted in Figure 2.1. The cache invalidator predictor (CIP) lies in between the flow that processes the stream of document that have been crawled and the cache. The CIP generates a *synopsis* or succinct summary of the document and then needs to locate quickly which cache entries are being invalidated by the incoming documents. In essence, this acts as a *reversed* search engine in which the documents are the queries and the role of the queries is played by the synopses,

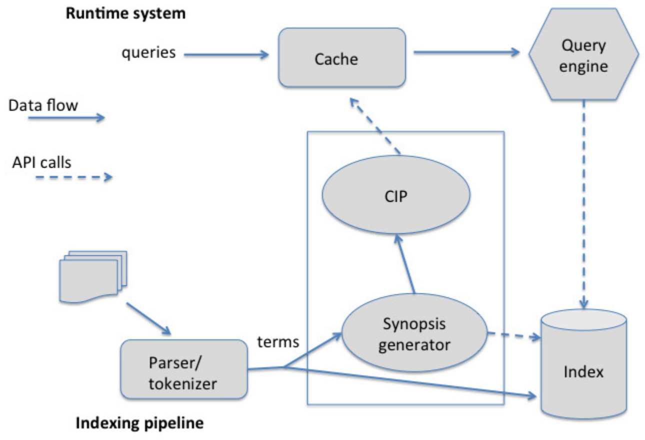


Figure 2.1: Architecture for selective cache entry invalidation, showing the components that conform the runtime system and indexing pipeline along with their dependencies, both API calls and data flow between the different components. The cache invalidator component accesses the terms generated through the indexing pipeline and calls the runtime cache in order to flag cache entries that should be marked as invalid.

similar to Pickens et al. (2010) who suggest this query index might be useful for query formulation, selection and feedback. They further defined metrics by which to measure the performance of these predictions, propose a realizing architecture for incorporating such predictors into search engines, and measure the performance of several prediction policies. Their results indicate that selective invalidation of cached search results can lower the number of queries invalidated unnecessarily by roughly 30% compared to a pure TTL schema, while returning results of equal freshness.

There are other ideas built upon having a time to live mechanism implemented in the cache. For instance, Alici et al. (2011) propose to maintain a separate timestamp for each document in the collection and posting list in the index. These timestamps indicate the point in time in which a particular posting became stale, as decided by a replacement policy. Whenever a user submitted query hits the cache, the engine compares the timestamps of the cached posting lists with the query timestamp in order to make a replacement decision. This approach is easy to implement and only performs slightly worse than simple CIPs. Given that it does not require generating a document synopsis each time the index is updated it also incur in a lesser number of operations for cache invalidation. It is not clear, though, what is the space overhead incurred in maintaining per posting timestamps and how this affects index compression and query performance runtimes (time-stamp checking in the presence of a cache hit). This work is extended by Alici et al. (2012) by setting TTL values individually per cached query. This stems from the rationale that too large TTL values will incur in serving stale results to the users, but using a too small value will diminish the efficiency gains of having a result cache.

Adaptive TTL values stem from the rationale that the results of different queries will remain fresh for different time periods. As we discussed previously, top results of queries looking for trending news or for results coming from highly dynamic sources (social media sites, for instance), will change very frequently, even in minutes (or seconds). However, user queries looking for historical facts or atemporal topics might remain fresh days or weeks. This is also affected by query seasonality patterns, in which queries that refer to some events may show temporal variation. For example, consider a query that refers to a particular event with some well-defined occurrence pattern (a sports competition, a TV show). The results for those queries will remain mostly unchanged throughout *non-peak* periods (when the event is not happening) but will have a high change rate whenever the events are taking place.

The solution to assign adaptive TTL values involve using a machine learning model that exploits features coming from both queries and results, such as number of query terms, query frequency, number of query results, among others. It still remains a challenge to devise a fully adaptive system that is able to account for peaks and changes in both the document and query stream, that is distributed, fully scalable and can handle the fast pace of modern crawlers, which return new documents at a sub-second pace.

2.4 Summary

Information retrieval systems that are aware of document changes need to adapt their basic data structures building blocks in order to be able to keep up with temporal variations content. The first step of any search engine, acquiring the document collection, is impacted when pages appear and disappear from a site. Crawlers must adapt their policies in order to optimize their resource usage and maximize the amount of *fresh* and new pages they discover and input into the indexing system. The index system must allow for incremental updates so that the new contents are readily available to be queried. Furthermore, if the retrieval systems is to provide search capabilities over different versions of documents it will require specialized indexing structures to cope with multiple time dependent document collection views. Finally, caching is one of the most impactful components from the efficiency perspective. If new relevant content is crawled and indexed for a substantial portion of queries, the cache needs to be aware that some of its entries are returning stale contents and invalidate or update them. To this end, we have reviewed several up-to-date solutions for the aforementioned problems that are able to scale up to the sizes of modern web collections.

3

Temporal Information Extraction

Both documents and queries contain implicit and explicit temporal information that can be utilized in the retrieval process. The same is also the case for other sources of textual information, including Internet encyclopedia like Wikipedia. The extracted information can be used as temporal profiles for different types of objects, namely, documents and queries, or a more specific data objects, i.e., entities and events.

The kind of temporal information that is easiest to extract are *temporal expressions*, which can be classified into explicit, implicit and relative temporal expressions as defined by Schilder and Habel (2003). An explicit temporal expression mentioned in a document can be mapped directly to a time point or interval, such as, dates or years on the Gregorian calendar. For example, “July 4, 2014” and “January 1, 2015” are explicit temporal expressions. An implicit temporal expression is given in a document as an imprecise time point or interval. For example, “Independence Day 2014” and “New Year’s Day 2015” are implicit expressions that can be mapped to “July 4, 2014” and “January 1, 2015” respectively. In addition, expressions such as “boston attack marathon” and “haiti earthquakes” can be understood as implicit temporal expressions, as they carry an inherent temporal nature that can be mapped to an explicit temporal point. A relative temporal expression occurring in a document can be resolved to a time point or interval using a time reference, either an explicit or implicit temporal expressions mentioned in a document or the publication date of the document itself. For example, the expressions “this Monday” and “next month” are relative expressions, where their exact dates can be determined using the publication date of the document. There are now several tools available for extracting temporal expressions and annotating documents, like the Stanford Temporal Tagger (Chang and Manning, 2012), TARSQI (Verhagen et al., 2005), and HeidelTime (Strötgen and Gertz, 2010).

In the rest of this section, we will describe the extraction of three main types of temporal information that requires more elaborate methods than the temporal expressions described above, namely: 1) the creation time of a document, 2) document focus time, and 3) temporal knowledge about entity and event evolution.

3.1 Document Creation Time

In order to leverage time for search, it can be beneficial to assign time of creation or published date to a document. The document creation time can also be needed in order to resolve relative temporal expressions that are relative to the publication time. However, it is challenging to find an accurate and trustworthy timestamp of a document. In many cases, the available time metadata of web documents are simply their last-modified dates or crawled time, which do not reflect the real creation dates (Nunes et al., 2007). Not being able to reliably detect creation dates not only severely affect temporal search effectiveness but also limit us from reasoning about date and time in other domains, such as, Topic Detection and Tracking (TDT) (Allan et al., 1998; He et al., 2007), and Emerging Trend Detection (ETD) (Berry, 2003).

Document dating is defined as the task of determining the time for non-timestamped documents, or, how to estimate the time of publication of document/contents or the time of topic of documents’ contents. Current approaches to document dating can be classified as either content-based or non-content-based methods.

3.1.1 Content-based

Content-based methods consider only the textual contents of documents, although an independent timestamped document collection is in general needed in order to create a model that can be used in the dating process. An important feature of these methods is that they can be used for any type of textual information.

The first approach was proposed by de Jong et al. (2005). The approach is based on temporal language models, which incorporates the time dimension into language modeling (Hiemstra, 2002). A language model M_D is estimated from a set of documents D, which is viewed as the probability distribution for generating a sequence of terms in a language. The probability of generating a sequence of terms can be computed by multiplying the probability of generating each term in the sequence (called a unigram language model), which is computed as:

$$P(w_1, w_2, w_3|M_D) = P(w_1|M_D) \cdot P(w_2|M_D) \cdot P(w_3|M_D). \tag{3.1}$$

In the *query likelihood model* (Zhai and Lafferty, 2004), a document d is ranked by the probability of a document d as the likelihood that it is generated by a query q, or $P(d|q)$. By applying Bayes' theorem, $P(d|q)$ can be computed as:

$$\begin{aligned} P(d|q) &= \frac{P(q|d) \cdot P(d)}{P(q)} \\ &\approx P(q|d), \end{aligned} \tag{3.2}$$

where $P(q)$ is the probability of a query q, and $P(d)$ is a document's prior probability. Both $P(q)$ and $P(d)$ are in general ignored from the calculation because they have the same values for all documents. The core of the *query likelihood model* is to compute $P(q|d)$ or the probability of generating q given the language model of d, or M_D. $P(q|d)$ can be computed using maximum likelihood estimation and the independence assumption as:

$$\begin{aligned} P(q|M_d) &= P(w_1, \ldots, w_{n_q}|M_d) \\ &= \prod_{i=1}^{n_q} P(w_i|M_d), \end{aligned} \tag{3.3}$$

where n_q is the number of terms in q. When estimating $P(w|d)$ for a query term w, we can encounter a problem of zero probabilities, which means that one or more terms in q may be absent from a document d. In order to avoid the problem, a smoothing technique can be applied in order to add a small (non-zero) probability to terms that are absent from a document. Such a small probability is generally taken from the background document collection. For each query term w, a smoothing technique is applied yielding the estimated probability $\hat{P}(w|d)$ of generating each query term w from d as:

$$\hat{P}(w|d) = \lambda \cdot P(w|M_d) + (1 - \lambda) \cdot P(w|M_C), \tag{3.4}$$

where the smoothing parameter $\lambda \in [0, 1]$. C is the background document collection. M_C is the language model generated from the background collection. In order to build temporal language models, a temporal corpus is needed. The temporal corpus can be any document collection where 1) the documents are timestamped with creation time, 2) covering a certain time period (at least the period of the queries collections), and 3) containing enough documents to make robust models. A good basis for such a corpus is a news archive. However, any corpus with similar characteristics can be employed, including non-English corpora for performing dating of non-English texts.

The temporal language model assigns a probability to a document $d_i = \{w_1, \ldots, w_n\}$ according to word usage statistics over time. The training corpus $C = \{d_1, \ldots, d_n\}$ is partitioned based on publication time, for example one partition per month or year, and for each partition. The approach employs a normalized log-likelihood ratio (NLLR) (Kraaij, 2005) for computing the similarity between two language models. Given a partitioned corpus C, it is possible to determine the time of a non-timestamped document d_i by comparing the language model of d_i with each corpus partition p_j using the following equation:

$$NLLR(d_i, p_j) = \sum_{w \in d_i} P(w|d_i) \times \log \frac{P(w|p_j)}{P(w|C)}. \tag{3.5}$$

$$P(w|d_i) = \frac{\mathit{tf}(w, d_i)}{\sum_{w' \in d_i} \mathit{tf}(w', d_i)}. \tag{3.6}$$

$$P(w|p_j) = \frac{\mathit{tf}(w, p_j)}{\sum_{w' \in p_j} \mathit{tf}(w', p_j)}. \tag{3.7}$$

$$P(w|C) = \frac{\mathit{tf}(w, C)}{\sum_{w' \in C} \mathit{tf}(w', C)}, \tag{3.8}$$

where $\mathit{tf}(w, d_i)$ is the frequency of a term w in a non-timestamped document d_i. $\mathit{tf}(w, p_j)$ is the frequency of w in a time partition p_j. $\mathit{tf}(w, C)$ is the frequency of w in the entire collection C. In other words, C is the background model estimated on the entire collection. The timestamp of the document is the time partition which maximizes the score according to the equation above. The intuition behind the described method is that given a document with unknown timestamp, it is possible to find the time interval that mostly overlaps in term usage with the document. For example, if the document contains the word "tsunami" and corpus statistic shows this word was very frequently used in 2004/2005, it is assumed that this time period is a good candidate for the document timestamp. When there are words with zero probability, a probability smoothing technique, such as, linear interpolation smoothing (Kraaij, 2005) or Dirichlet smoothing (Zhai and Lafferty, 2004), can be applied in order to assign some small (non-zero) probability to words absent from a time partition.

The approach of de Jong et al. (2005) was subsequently extended by (Kanhabua and Nørvåg, 2008). The most important improvement was the use of temporal entropy, motivated by the fact that some terms are more suitable than others for distinguishing *documents* from others, and term entropy can be utilized to select those terms. Similarly, some terms are better for distinguishing *between time periods*, and Kanhabua and Nørvåg introduce *temporal entropy* which is employed to modify the term weighs described above for the de Jong approach. They also study the impact of varying semantic-based preprocessing techniques, as well as propose a method taking into account what were the top gaining and declining queries on Google Zeitgeist (an aggregation of search queries for each year).

The time range covered by a content-based document dating approach depends on the training corpus. Kumar et al. (2011) study the use of different training sets for the de Jong approach, and in particular how a training set consisting of Wikipedia biographies spanning 3800 B.C. to 2010 A.D. can be used to predict publication date of older texts.

A basic approach to solving the document dating task is to use classifiers. Chambers (2012) propose a discriminative classifier with new features extracted from the text's time expressions and a second model that learns probabilistic constraints between time expressions and the unknown document time. This approach outperforms the methods proposed by de Jong et al. (2005) and Kanhabua and Nørvåg (2008), however it is restricted to predicting year only. Ge et al. (2013) subsequently extended the work of Chambers by combining the use of classifier with event-based time label propagation, thus also utilizing relative temporal relations between documents and events.

A significantly different approach to the aforementioned methods is to use the burstiness of terms for estimating timestamps (Kotsakos et al., 2014). Previous methods can only report time based on a pre-defined granularity that reflects the segmentation of the reference corpus. However, this approach can also report non-fixed intervals of application-defined length l. It is based on the intuition that similar documents are more likely to discuss similar events and hence being created closer in time, and that the burst intervals of significant terms (for example selected based on *tf-idf*) in those documents having high degree of overlap. The document dating process is performed by first finding the documents most similar to the document to be dated. Second, a weight is assigned to each of the related documents based on the overlap of burst intervals of common terms between the relevant document and the document to be dated. Finally, each publication date along the timeline is assigned the sum of the weights of documents published at that time, and the result interval is chosen as the time interval of length l having maximum sum of weights. Based on the experimental evaluation, this is the current state-of-the art approach to content-based document dating. Please refer to (Kotsakos et al., 2014) for the detailed description of the aforementioned approach.

In addition to the above language-independent approaches, there are also works taking into account language and/or domain in order to improve accuracy of the dating process. Garcia-Fernandez et al. (2011) describe a system for determining publication date of old French newspaper articles, combining supervised and unsupervised methods. The most important contribution is the use of Google Books N-grams (Goldberg and Orwant, 2013) in order to automatically identify neologisms and archaisms. Tilahun et al. (2012) focus on dating medieval English charters, and demonstrate that the use of techniques similar to those proposed in (de Jong et al., 2005; Kanhabua and Nørvåg, 2008), but applied on shingles (n-grams) instead of terms, perform well on that task.

3.1.2 Non-Content-based

Non-content-based document dating uses information external to the documents. The main advantage is not requiring a reference corpus, but put strong constraints on the availability and accuracy of other information making it less useful in practice. Jatowt et al. (2007) present an approach that searches past versions of a page in a web archive in order to detect when it was created. An important shortcoming of this method is the need for having the page archived and in sufficient granularity to detect when changes happened, and also the cost of retrieving pages from the archive.

Nunes et al. (2007) propose to use time information from neighbors of a document, such as 1) documents containing links to the non-timestamped document (incoming links), 2) documents pointed to the non-timestamped document (outgoing links, as also suggested previously by Hauff and Azzopardi (2005)) and 3) the media assets (e.g., images) associated with the non-timestamped document. They compute the average of last-modified dates extracted from neighbor documents and use it as the time for the non-timestamped document. This approach is essentially limited to linked documents like web pages, and accuracy will depend on the accuracy of the time of dating of the neighbors. SalahEldeen and Nelson (2013) extend the previous ideas by using external sources to determine when an URL was first used, for example, when a shortlink was first created using Bitly[1]. The drawback of this method is the lack of resources covering further into the past than the most recent years, and it also cannot determine the age of the actual contents on the page, only when the web page represented by the URL first appeared.

3.2 Document Focus Time

While document dating aims at determining the creating date of a document, often it is also useful to determine the date or time period which the contents refers to. We will in this section describe how the task of estimating *document focus time* can be addressed. It is important to notice that some documents are atemporal, so that mapping documents to certain time periods only makes sense for those having temporal characteristic.

A basic approach for estimating focus time is to base the decision on extracted temporal expressions. Extracting temporal expressions is typically performed by first identifying time entities in the text, and then performing temporal reference resolution in order to convert the relative time entities into absolute time. The temporal reference resolution is typically rule based, for example, Mani and Wilson (2000) solve the task by using a mix of hand-crafted and machine-learnt rules, similar approaches can also be found in other papers, e.g., (Llidó et al., 2001). The focus time can then be determined, for example, by ranking the temporal expressions and considering the most highly ranked as the focus time (Strötgen et al., 2012).

Using extracted temporal expressions is not always sufficiently accurate. Jatowt et al. (2013) propose an alternative approach where document focus time is estimated based on what year the terms in the document are most strongly associated with. In order to determine this

[1] Bitly is a URL shortening service (`https://bitly.com/`).

association, a corpus of news articles is used. When estimating focus time, (1) temporal entropy – a measure of how discriminative a term is for distinguishing between time periods, cf. (Kanhabua and Nørvåg, 2008), and (2) temporal kurtosis (the less peakedness in the distribution of a term over time means the stronger association to particular time periods, see Section 4.1.2), are used to determine which terms in the document should be used. The document focus time is then selected as the time that maximizes the document-time association based on those selected terms.

3.3 Entity and Event Evolution

It has been observed that entities and events are changing over time, which comprise both the evolution of the entities and events themselves and the evolution of their representations. In general, such evolutions can reflect in two main issues: *terminology change* and *context change*. Terminology evolution and the mapping between current to historic terms will be explained in more detail in the context of time-aware query reformulation in Section 4.3.

Common changes in context would include changes in organizational roles, personal relationships, and world-related knowledge, e.g., geo-political changes. Moreover, we can anticipate changes occurring in different periods of life or in workplace and societal settings. This is partially due to the evolving natures of the social Web and collective attentions (Mazeika et al., 2011). Participative content generation and sharing in Web 2.0 offer new rich data sources for a large scale analysis of patterns in human and especially collective attentions as a crowd phenomenon. The social negotiation and construction processes, e.g., are reflected by early editing activities on pages referring to real-world events (Ferron and Massa, 2012), as well as by discussion forums for each page on Wikipedia or talk pages (Pentzold, 2009). Previous works exploit edit history (Georgescu et al., 2013) and article view statistics (Ciglan and Nørvåg, 2010) for detecting events and entities related to the events, which provide promising results towards generating entity-specific news tickers and timelines.

The study of entity and event evolution comprehends a wide range of IR applications, for example, web archiving, temporal search and longitudinal analytics (Alonso et al., 2011b). Incorporating information about evolving entities and events is useful to enhance search results.

Kanhabua and Nejdl (2014) point out that it is important to adequately deal with such evolutions, for example, to *track* and *detect* changing properties of entities and events over time. A key challenge to tackle this problem includes a careful identification of selected context dimensions, which information or features (e.g., properties of the entities) should be captured and affected by the respective evolution. This process of selecting adequate features is expected to create varying results depending upon the type of information and the use case under consideration. It is expected to be able to identify some core time travel representations depending on types of evolution supported by the designed and implemented methods. To this end, there is the need for tools to deal with evolving data relevant to observed information and adapt context.

Entity and event evolution, i.e., detecting and extracting changing properties of entities over time, has been proposed using different methods such as rule-based techniques (Ernst-Gerlach and Fuhr, 2007), temporal association rule mining (Kaluarachchi et al., 2010a), and mining Wikipedia (Kanhabua and Nørvåg, 2010b). In addition, an extension of the YAGO knowledge base, called YAGO2 (Hoffart et al., 2013), was created by incorporating the temporal facts about entities, which are extracted from semi-structured contents like Wikipedia infoboxes.

Kanhabua and Nejdl (2014) extend the previous work (Kanhabua and Nørvåg, 2010b) for discovering entity evolution using *temporal anchor texts*, which constitute anchor texts and corresponding time snapshots mined from the edit history of Wikipedia. After extracting anchor texts for each snapshot, the weights of aggregated anchor texts will be computed and used for ranking them by importance with respect to a target entity. For a given entity e, the set of temporal anchor texts $A_{e,t}$ contains all the unique anchor texts of e's incoming links at time t. Each anchor text a is associated to a weighting function $f(a, e, t)$, which can be calculated as the count of inlink pages to e or using a better estimation taking into account the relationship among sites or domains, i.e., considering inlink pages from the same site/domain as *internal links* and those from different domains as external links (Dai and Davison, 2010b; Dou et al., 2009).

Table 3.1 depicts time-evolving information associated to some selected entities including Barack Obama, Pope Benedict XVI, and Microsoft Windows. As observed, temporal anchor texts is useful towards capturing entity evolution information, namely, the changing of names or political roles, as well as trends (e.g., popular songs or current shows/performance of artists) and evolving context (e.g., new products, software versions or related events).

Table 3.1: Examples of entities with evolving information captured by temporal anchor texts.

Entity	Time	Anchor Text	Weight
Barack Obama	200807	Senator Barack Obama	2.71E-07
Barack Obama	200807	Illinois Senator Barack Obama	3.39E-08
Barack Obama	200807	U.S. Sen. Barack Obama	1.69E-08
Barack Obama	200903	President Barack Obama	7.59E-07
Pope Benedict XVI	200506	Joseph Cardinal Ratzinger	1.03E-05
Pope Benedict XVI	200506	the new pope	5.70E-07
Microsoft Windows	200210	Windows CE	1.60E-05
Microsoft Windows	200212	Windows XP	9.44E-06
Microsoft Windows	200601	Windows 2000, XP, Server 2003	2.85E-07
Microsoft Windows	200607	Windows Me, 2000, XP, Vista	1.94E-07

3.4 Summary

Documents, queries and other textual sources contain temporal information that can be utilized in the information retrieval process. We have provided an extensive overview of fundamental tasks of temporal information extraction. Determining document creation time and document focus time is useful not only in the context of the documents themselves, but can also be exploited to gain information from queries and be employed for ranking. In additional to extracting information about *time*, entity and event evolution can also be extracted from temporal documents. Such information has many applications, knowledge of entity evolution can for example be used as part of a query expansion in order to increase recall, while entity evolution can be used to enhance search results. In the subsequent sections, we will see in more detail how the extracted information can be applied.

4

Temporal Query Analysis

This section presents research problems related to temporal queries and provides an overview of existing works in this area. The structure of this section consists of three main parts, i.e., temporal query intent (Section 4.1), dynamic query subtopics (Section 4.2), and time-aware query enhancement (Section 4.3).

Understanding search intent behind a user's query is the first step before applying a suitable retrieval and ranking model. In the temporal search realm, a system should be able to detect a query that contains an underlying temporal information need, so-called a *temporal query* (Joho et al., 2013). We categorize temporal queries into two main categories. The first class of temporal queries can be observed through *temporal patterns* of user search behavior in query logs, whereas the second class of temporal queries may not exhibit such temporal patterns in a query stream, but contain underlying temporal information needs.

Examples of first-class queries are: *sporadic or spiky* queries, e.g., breaking news, celebrities, and short-span events, *periodic or seasonal* queries, such as, annual events, and *trends* consisting of anticipated events and ongoing events. For the second class, temporal queries are either (1) those with temporal criteria explicitly provided (i.e., containing temporal expressions), or (2) those with no explicit temporal criteria. An example of a query with time explicitly provided is World Cup 2014, while Brazil World Cup is a query without temporal criteria provided, implicitly provided, and it which can refer to either the most recent FIFA World Cup event in 2014 or the historical event in 1950.

Figure 4.1 illustrates example queries with various temporal characteristics. In the case of a temporal query referring to one particular event, it can contain dynamic subtopics, where its relevant subtopics are highly time-dependent. For example, when issuing the query kentucky derby in April, search intents are likely to be about hats and food referring to the Kentucky Derby Festival, which occurs two weeks before a stakes race. Presumably, at the end of May, other related searches like result or winner should be more relevant to than pre-event aspects. As another example, the query NCAA tournament[1]. We depict relevant subtopics at three different time points (i.e., *March 14*, *March 18* and *April 1*), where the change in temporal aspects of a query reflects the ongoing event. In the beginning of the event, *pre-event* aspects, such as, march madness and a sub-event like women's basketball tournament bracket can be regarded as trend in search streams. At the later stage, post-event and late-event aspects have gained their popularity, for instance, in the case of ncaa basketball results for *post-event* and ncaa final four for *late-event*.

A temporal query may be associated to different events, which is regarded as *topically ambiguous*. An example of such *multi-faceted temporal query* (Whiting et al., 2013) is U.S. Open, which can relate to one of these two sport events: (1) the annual open golf tournament of the United States typically held in mid-June, or (2) the fourth and final tennis major that comprises the Grand Slam held annually in late August and early September. In addition, a temporal query is seen as *temporally ambiguous* (Jones and Diaz, 2007), which can be associated to multiple events occurred in different periods in time. For example, FIFA World Cup and tsunami can refer to many events, for both recurring and non-recurring ones, in the past that the issued queries can refer to.

[1] A sports competition held annually by the National Collegiate Athletic Association.

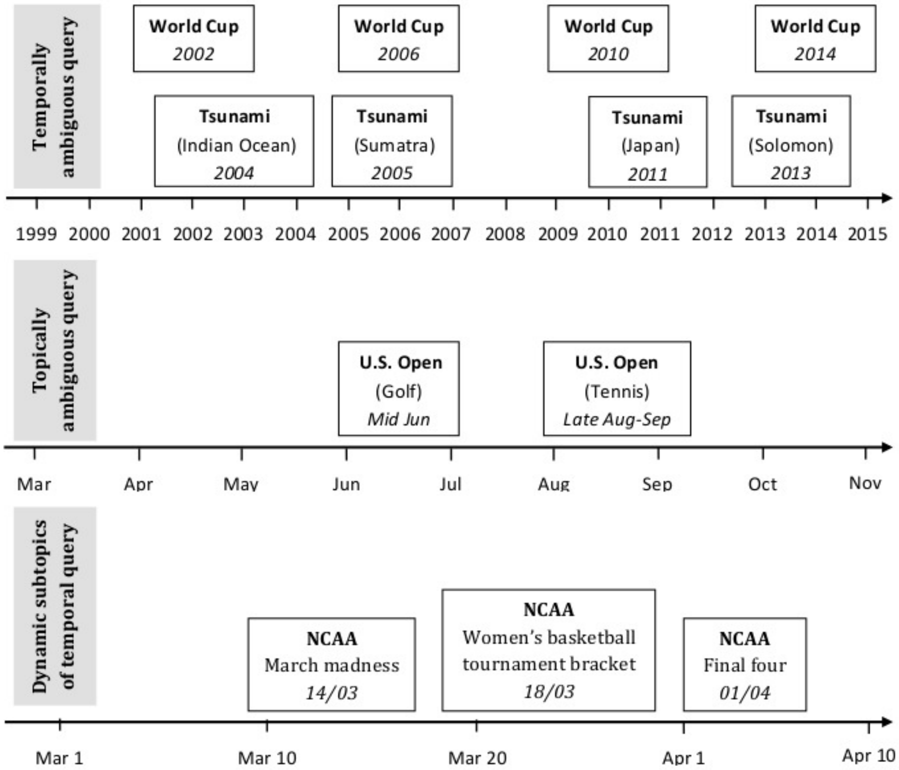


Figure 4.1: Variances of temporal queries and their dynamic characteristics.

The rest of this section is organized as follows. Section 4.1 describes current approaches to understanding temporal query intent, as well as methods for detecting and predicting time-sensitive queries, such as, spiky, periodic or seasonal queries. In addition, we will explain how to determine the underlying temporal information needs for a given temporal query. In Section 4.2, we will address the problem of *multi-faceted temporal queries* and its impact on search result diversification, and outline recent works dealing with *topically ambiguous temporal queries*. Finally, we give an overview of query enhancement techniques applied to temporal queries for increasing search experience in Section 4.3.

4.1 Temporal Query Intent

Existing methods on determining temporal query intent can be classified into two main groups based on the data sources used, namely, (1) mining temporal patterns in query streams, and (2) determining temporal intent from top-k search results.

4.1.1 Mining Temporal Patterns in Query Streams

By analyzing changes in historical query logs, we can observe a sort of signal provided a basic understanding for temporal querying behaviors using standard statistic methods as done in (Beitzel et al., 2004, 2007; Richardson, 2008; Kulkarni et al., 2011). Moving beyond analyzing popularity changes, we present several works on automatically detecting and classifying temporal queries (Vlachos et al., 2004; Jones and Diaz, 2007; Parikh and Sundaresan, 2008; Kulkarni et al., 2011). Finally, we explain methods for modeling and predicting query popularity using time series techniques (Radinsky et al., 2012; Shokouhi, 2011).

Analyzing Changes in Query Popularity

Beitzel et al. (2004, 2007) analyze changes in query popularity by focusing on queries that are repeated often, rather than those in the long tail. They manually assign a sample of 600 queries to the pre-defined, topical categories and study changes in popularity of the categorized queries over time on daily, weekly, and monthly scales.

A direct comparison of the percentages of total query volume matching a selected group of category lists is difficult due to the differences in scale. To overcome this problem they measure relative popularity changes or fluctuations over time of each query category, which can be computed as follows.

$$KL(P(q|t)||P(q|c,t)) = \sum_{q} P(q|t) \times \log \frac{P(q|t)}{P(q|c,t)}, \tag{4.1}$$

where $KL(P(q|t)||P(q|c,t))$ is the KL-divergence (Kullback and Leibler, 1951), or a non-symmetric measure of the difference between two probability distributions, i.e., the likelihood $P(q|t)$ of receiving any query q at a particular time t and the likelihood $P(q|c,t)$ of receiving q in a particular category c. In their paper, they show the divergences (popularity fluctuations) over hours in a day among different query categories. Although the categories that cover the largest portions of the query stream also have the most relative popularity fluctuation, this correlation does not hold throughout all categories. Note that, two topics *music* and *movies* are examined independently from other queries in the category *entertainment*. The results show that the trends of query popularity of particular topics differs both from the query stream as a whole and from other categories. Additionally, certain categories of queries trend differently over varying periods.

Richardson (2008) determines changes in popularity by examining whether the distribution of queries within a session is significantly different from the general population distribution. More precisely, Richardson measures the query distributions, among users who searched for a query vs. all users, using KL-divergence. This solely studies the change in interests over time of a few specific types of queries with sub-sequential related searches spanning over days or months, e.g., migraine (*related searches*: tea, coffee, caffeine, and magnesium), mortgage (*related searches*: lending, realtor, and loan), interview (*related searches*: resume, tax, and moving), and dating (*related searches*: proposal, engagement ring, and wedding planning). The analysis results have twofold remarks. First, learning from common similar trends in histories is useful, e.g., relationship between medical condition and potential causes. Second, long-term history could be more effective for generating topic hierarchies than short-term history.

To summarize, a metric or distance based on a probability and information theory such as KL-divergence can intuitively capture the temporal evolution for better understanding and a analyzing changes in querying behavior. However, the limitation of the aforementioned method is that it is unsuitable for modeling changes in query popularity towards query intent detection and prediction.

Detecting and Categorizing Temporal Queries

Vlachos et al. (2004) are among the first to apply a burst detection method (Kleinberg, 2003) for identifying temporal queries based on three classes of temporal patterns: 1) *periodic* queries, for example, an event happening in a weekly basis like cinema or nordstrom, 2) *seasonal* queries recurring monthly or yearly, for instance, full moon, Easter, or Halloween, and 3) *large peak* queries for rare or unseen events or breaking news, e.g., dudley moore (during the day that the famous British actor died). An essence in their proposed approach is an efficient and effective method for identification of bursts (long or short-term) extracted from a query sequence, and support of *query-by-burst* on the database of time series. This method enables the discovery of semantically similar queries by identifying queries with similar temporal patterns, called "query-by-burst".

Parikh and Sundaresan (2008) use a burst detection method for classifying temporal queries based on the shape and duration of bursts, which can indicate different causalities caused by external events, such as, celebrity news, fraud, or an emerging trend. The query streams are modeled as a mixture of queries using a binomial distribution. A key feature they proposed is *incremental burst detection* capable of dealing with a high volume of query streams, and detect bursts in an incremental and scalable way as a new query arrives. This technique of near real-time detection of these patterns aid in building applications that take an immediate action like fraud detection or product merchandising.

To get a deeper understanding of triggering events, the detected bursts are manually classified based upon their shapes (spanning over an event period) and duration into four classes by matching shapes of peaks in the Canadian Rockies mountain including (1) *Matterhorns*: the most interesting with sharp narrow spikes referring to products with a pre-launch hype, e.g., iphone and queries related to sports stars, celebrities and movie launches, e.g., the simpsons, (2) *Cuestas*: bursty patterns mostly with broad peaks or dual peaks, e.g., products with an initial limited release and a follow-up release, or players performing well on multiple occasions, e.g., lebron james and alex rodriguez, (3) *Dogtooths*: various peaks and distribution having some elements of uniformity over time, e.g., perfume bottles, longaberger baskets and hanna andersson, and (4) *Hogbacks*: steadily increased or evolving over time like a trend, e.g., products penetrate the market and their popularity steadily grows after launch, e.g., Treo, T-Mobile products and launch of some toys like Transformers.

Kulkarni et al. (2011) propose a classification scheme for temporal queries by observing changes to query popularity over time, as well as changes to other components, i.e., document content and relevance. Queries with similar patterns could be classified into the same temporal group. They characterize the distribution of query popularity along four dimensions: 1) the number of spikes, 2) the shape of spikes, 3) query periodicity, and 4) an overall trend in popularity. This classification scheme was iteratively developed using an affinity diagramming technique (Beyer and Holtzblatt, 1998). Kulkarni et al. (2011) categorize 100 queries using a two-phase process by first analyzing all of the query popularity shapes to develop a categorization scheme, and then re-analyzing all queries to classify them. Table 4.1 summarizes the features used for categorizing queries based on their popularity changes.

In addition to popularity change features, Kulkarni et al. (2011) measure result content changes as an indicator that a particular information need evolves, new relevant pages are created, old pages die, and others are updated. In their work, two types of content change features were used: (1) query-dependent, i.e., the term frequency (TF) of the query on page over the studied period, and (2) query-independent, i.e., using the Dice coefficient, as a measure of how much the overall page content changes over time. The Dice coefficient measures the textual overlap among different document versions, which can be computed as:

$$Dice(W_i, W_j) = 2\frac{W_i \cap W_j}{|W_i| + |W_j|}, \tag{4.2}$$

where W_i and W_j are sets of words for the document at time i and j, respectively. This metric is also used by Adar et al. (2009) for measuring the temporal dynamics of web content. The relationship between the degree of changes in query popularity and result content change reveals some interesting insights, which are summarized as follows.

Table 4.1: Features for measuring changes in query popularity (Kulkarni et al., 2011).

Feature	Value	Description
Number of spikes	0, 1 or multiple	A spike occurs when there is a sudden increase followed by a corresponding decrease in query popularity. This attribute captures the number of times such a change occurs during the 10 week study period.
Periodicity	yes, no	A consistent repetitive pattern of spikes during the time-frame of the study is considered a periodicity.
Shape	wedge, sail, castle	When a query spikes, the spike can have one of the following shapes. (1) *Wedge*: The popularity rises over time at the same rate that it later falls off. (2) *Castle*: The popularity changes (rises or drops), and stays at the new level for a relatively long period of time (roughly a few weeks or more). (3) *Sail*: The query popularity rises somewhat slowly (roughly over a week) and then dramatically drops off over a short period of time (roughly 1-2 days) or conversely rises sharply but drops off slowly after reaching the peak popularity.
Trend	up, down, flat, up-down	The query popularity can exhibit an overall increase or decrease over the duration of the study. The trend attribute encodes this property of popularity.

- *High popularity change, Low content change.* Some queries exhibited relatively low content change despite spiking significantly in popularity, e.g., annual events like Easter (hard boiled eggs) or April 15 US tax day (taxes online).

- *High popularity change, High content change.* The query mlb displayed more than twice as much change in both the measures, presumably because the baseball season began during the studied period, and had a lot of change to associated pages.

- *Periodic/aperiodic, High content change.* There is a strong significant trend that periodic queries change more in overall content than aperiodic queries. Periodic queries were often related to popular television shows or sports events associated with highly dynamic content, whereas aperiodic queries that underwent substantial content change tended to be celebrity queries.

- *Up-down popularity.* Queries for which the overall popularity was an up-down trend (april fool, cma awards) were associated with low content change while those with the overall popularity either rose (glee, justin bieber) or stayed flat (adam lambert, giada de laurentis) exhibited higher change in content.

Modeling and Predicting Changes in Query Popularity

We present temporal query modeling methods proposed in previous works (Radinsky et al., 2012; Shokouhi, 2011), which make use of different time-series analysis techniques (Hamilton, 1994) for capturing the temporal dynamics of web search behavior. The most basic technique is to recognize the seasonal pattern of temporal queries. Shokouhi (2011) studies how periodic an observed querying behavior over time is, and conducts a time-series analysis for determining seasonal queries. To extract the seasonality component, time series data can be decomposed by different statistical techniques (Cleveland et al., 1990; Holt, 2004), called a time-series decomposition process. Given a time series $Y = \{y_1, ..., y_N\}$ where N is the total number of observed values, Holt-Winters adaptive exponential smoothing (Holt, 2004) can be applied to construct the statistical decomposition of time series input, which can be either the aggregated query volume or the distribution of top-k retrieved documents over time. The output consists of three main components: level L, trend T and seasonality S. The seasonality component S is important for estimating the seasonal characteristic (a pattern that repeats itself with a known periodicity, such as, every week or annually) of a given time series Y. The final score can be calculated using a similarity metric, i.e., the cosine similarity between the original time series Y to its derived seasonality component S as follows:

$$Seasonality(Y, S) = \frac{\vec{Y}\vec{S}}{\parallel \vec{Y} \parallel \cdot \parallel \vec{S} \parallel}. \quad (4.3)$$

Figure 4.2 depicts an example of time series decomposition for a time series of search results for the query easter. Amodeo et al. (2011b) use features derived from the time series decomposition to classify queries into different temporal classes and also for predicting future events of importance related to the query topic. However, leveraging a time-series decomposition technique alone might not be sufficient to model and predict dynamic search behavior. On the other hand, Radinsky et al. (2012) propose a temporal model that can be used to capture and predict time-varying user behavior by employing several features from physics and signal processing including global and local trends, periodicities, and surprises. A global trend is modeled using a time-series smoothing technique, namely, the simple Holt-Winters model for producing an exponentially decaying average of all past examples, thus giving higher weights to more recent events. In this case, the trend represents the long-term direction of the time series. A local trend, and a periodic or seasonal component are computed using the time series decomposition technique presented above.

In addition to the aforementioned temporal querying behavior, Radinsky et al. (2012) also model *a surprise* in query streams. The intuition is that in a real-world setting, a model itself changes over time caused by external disturbances, which might have a strong effect on the forecast and parameter estimation of the (fitted) model. In this case, they augmented the standard Holt-Winters model with a

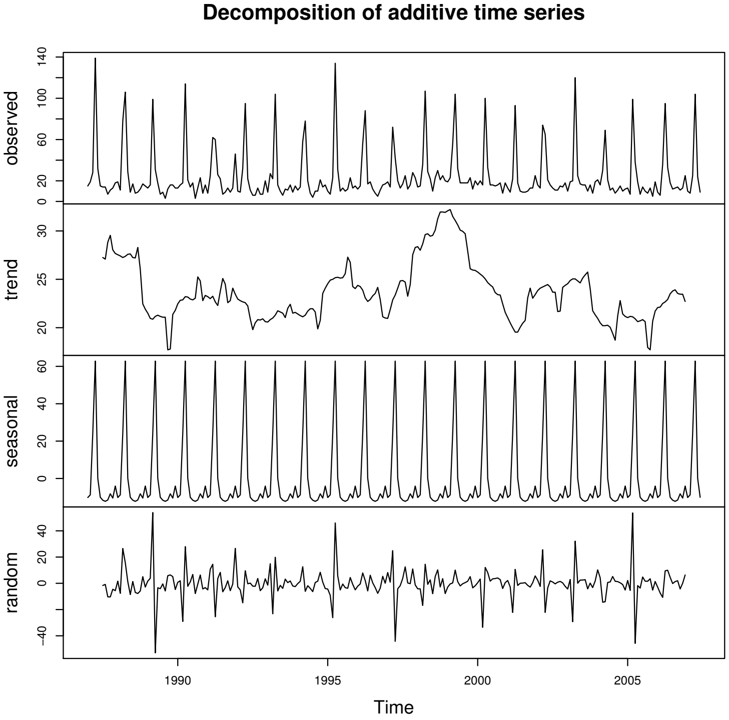


Figure 4.2: Time series decomposition of top-k search results of the query easter.

surprise measurement. Specifically, a surprise represents an unplanned event happened when there is a significant error in the residuals of a temporal model, i.e., by computing the sum of squared errors of prediction (SSE). The score implies how *unplanned* is the time series at a given time point. Given a time series $Y = \{y_1, ..., y_N\}$, its predicted values $\hat{Y} = \{\hat{y}_1, ..., \hat{y}_N\}$ can be estimated using a standard linear simple regression model. The surprise score is calculated as:

$$SSE(Y, \hat{Y}) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{4.4}$$

To this end, the set of derived features consists of aggregate features of the time series, shape feature of the time series, and other domain-specific features such as the query class. They employ the proposed features to train temporal query models and employ machine learning to infer which of the trained models is most appropriate for prediction dynamic search behavior. For this purpose, they propose a new learning algorithm, called dynamics model learner Radinsky et al. (2012), for learning from multiple objects with historical data.

4.1.2 Determining Temporal Intent from Top-k Results

When no temporal querying behaviors (or query popularities) information is available, determining temporal query intent can be performed by analyzing top-k retrieved search results.

Learning to Classify Temporal Queries

Jones and Diaz (2007) define *temporal profiles* as the temporalities or temporal characteristics of queries, where such information can be used for classifying a query into predefined classes. Query classification proposed by Jones and Diaz (2007) comprises three types of queries, namely, (1) *temporally unambiguous* taking place at a specific period in time, (2) *temporally ambiguous* taking place during one of several possible episodes, and (3) *atemporal* taking place at any time. Those classes are identified by temporal patterns of documents retrieved in response to queries. It is important to note that temporal query classification is a difficult task as reported in a recent work by Kanhabua et al. (2015), where most of previous works have encountered similar challenges.

While the temporal profile of a query can be determined by mining query logs, a long-term history of search logs might not be easily accessible. When issuing an event-related query in a web search engine, a set of retrieved documents can exhibit a particular characteristic, i.e., burstiness, in its temporal distribution. This is due to many documents are published around the time period of an event underlying a given query. Leveraging a temporal profile based on top-k search results, e.g., by selecting the most descriptive terms from different time periods as done in (Kanhabua and Nørvåg, 2010a), can provide a better query representation, or improve a query modeling approach. They introduce a language modeling approach for capturing the temporal nature of a query using a probability distribution $P(t|q)$, where t is the day relevant to the searcher. They assume the top-k retrieved documents D_q for a given query q as a proxy for the set of relevant documents, and weight each by the estimated relevance scores obtained from the retrieval model.

$$P(t|q) = \sum_{d \in D_q} P(t|d) \frac{P(q|d)}{\sum_{d' \in D_q} P(q|d')} \ , \tag{4.5}$$

where $P(t|q)$ is the probability of generating a publication date t given q and D_q is top-k documents retrieved with respect to q. $P(t|q)$ is defined using *relevance language modeling* presented by Lavrenko and Croft (2001), that is, the top-k retrieved documents D_q are considered and weighed according to the document's probability of relevance, i.e, $P(q|d)$. In other words, $P(q|d)$ is a retrieval score of d for a particular ranking model.

$$P(t|d) = \begin{cases} 0 & \text{if } \mathit{PubTime}(d) \neq t, \\ 1 & \text{if } \mathit{PubTime}(d) = t. \end{cases} \tag{4.6}$$

The probability distribution might not be available for all dates due to missing result documents for a specific date. Thus, the overall distribution should be smoothing with an appropriate background probability to handle potential irregularities in the collection distribution over time. Background smoothing replaces zero probability events with a very small probability, where we have to assign a very small likelihood of a topic being discussed on days where we have no explicit evidence. The final probability $\widetilde{P}(t|q)$ can be computed as follows:

$$\widetilde{P}(t|q) = \lambda P(t|q) + (1 - \lambda) P(t|C) \ , \tag{4.7}$$

where λ is a smoothing parameter. More precisely, $P(t|C)$ is the probability of a publication date t in the collection C calculated as:

$$P(t|C) = \frac{1}{|C|} \sum_{d \in C} P(t|d) \ . \tag{4.8}$$

Note that, some smoothing methods (e.g., moving average) other than the one described above can be an alternative as well.

It has been shown in (Jones and Diaz, 2007) that the temporal profiles of two queries poaching and deer, which are *atemporal*, appear relatively uniform. For temporally unambiguous queries like earthquake in armenia and matrix, their temporal profiles suggest that the queries refer to specific points or periods in time. In this context, a query is only temporally unambiguous with respect to a specific collection. On the other hand, temporally ambiguous queries, such as, hostage taking and NBA basketball playoffs, have less distinct temporal profiles than the temporally unambiguous queries, but nonetheless are distributed quite differently from the temporal profile of the entire collection.

The next step is to leverage the temporal profiles of queries and other temporal features (e.g., temporal KL-divergence, autocorrelation, kurtosis, etc.) derived from analyzing the top-k retrieved documents for classifying queries into the predefined classes. We present the studied temporal features proposed in (Jones and Diaz, 2007) as follows.

Temporal KL-divergence measures the difference between temporal distributions of top-k documents and the collection computed as:

$$\textit{Temporal-KL}(D_q||C) = \sum_{t \in T} P(t|q) \log \frac{P(t|q)}{P(t|T)} , \tag{4.9}$$

where T is the set of all publication dates in the collection C. $P(t|T)$ is the probability of a publication date t in the collection. $P(t|q)$ is the probability of generating a publication date t given q and D_q is top-k documents retrieved with respect to q. $P(t|q)$ is a retrieval score of d for a particular ranking model, e.g., relevance language modeling proposed by Lavrenko and Croft (2001). Thus, the top-k retrieved documents D_q are ranked according to the document's probability of relevance $P(q|d)$.

Autocorrelation is the cross correlation of a signal with itself or the correlation between its own past and future values at different times. Autocorrelation can be exploited for predicting query popularity because future values depend on current and past values. In addition, it can be used for identifying repeating patterns, such as, the presence of a periodic signal obscured by noise, or finding the missing fundamental frequency in a signal implied by its harmonic frequencies (Box and Jenkins, 1990). When an event contains a strong inter-day dependency, the autocorrelation value will be high. Given observed time series values $\{y_1, ..., y_N\}$ and its mean $\bar{y}$, autocorrelation is the similarity between observations as a function of the time lag l between them.

$$Autocorrelation(Y, l) = \frac{\sum_{i=1}^{N-k} (y_i - \bar{y})(y_{i+l} - \bar{y})}{\sum_{i=1}^{N} (y_i - \bar{y})^2}. \tag{4.10}$$

Jones and Diaz (2007) consider only the autocorrelation at the one-time unit lag only ($l = 1$), i.e., shifting the second time series by one day. This is called the first-order autocorrelation of the first $N - 1$ observations $\{y_1, ..., y_{N-1}\}$ and the next $N-1$ observations $\{y_2, ..., y_N\}$.

Kurtosis is a basic statistic method to measure the *peakedness* or *skewness* of a probability distribution, i.e., considering how much of the probability distribution is contained in the peaks, and how much in the low-probability regions. Kurtosis is calculated as the average deviations of the time series elements to the forth power divided by the standard deviation to the forth power. Intuitively, kurtosis can capture sporadic or spiky events, such as, breaking news, celebrities, and short-span events. It quantifies how much of the probability distribution is contained in the peaks, and how much in the low-probability regions. Kurtosis is a popular statistical measure of 'peakedness', which is calculated as the ratio of the fourth moment and variance squared, which can be calculated as follows:

$$\begin{aligned} Kurtosis(Y) &= \frac{\mu_4}{(\sigma^2)^2} \\ &= N \frac{\sum_{i=1}^{N} (y_i - \bar{y})^4}{(\sum_{i=1}^{N} (y_i - \bar{y})^2)^2}. \end{aligned} \tag{4.11}$$

where μ_4 is the fourth moment about the mean and σ is the standard deviation. It has been observed that temporal features alone could not achieve high accuracy for query classification. A feature based on an analysis of the top-k retrieved documents, such as, the content clarity originally proposed by Cronen-Townsend et al. (2002) can also be employed for improving the performance of temporal query classification.

Determining Relevant Time for Queries

Kanhabua and Nørvåg (2010a) propose three approaches to determining the time of queries when no temporal criteria are provided, so-called *dating queries*, which is an analogous task for determining *relevant time* associated to a document (see Section 3). The first two approaches to dating queries use the temporal language models (see Section 3.1.1), and the last approach uses no language models. The first one performs dating queries using keywords only. The second one takes into account the fact that in general queries are short, and aims at solving this problem with a technique inspired by pseudo-relevance feedback (PRF) that uses the *top-k* retrieved documents in dating queries. The third one also uses the *top-k* retrieved documents by PRF and assumes their creation dates as temporal profiles of queries. All approaches will return a set of determined time intervals and their weights, which can be used for enhancing subsequent IR processes, such as, incorporate temporal relevance into a ranking model in order to improve the retrieval effectiveness.

A basic technique for query dating is based on using keywords only, and it is described formally in Algorithm 1. The first step is to build temporal language models T_{LM} from the temporal document corpus (line 5), which essentially is the statistics of word usage (raw frequencies) in all time intervals, which are partitioned with respect to the selected time granularity g. Creating the temporal language models (basically aggregating statistics grouped on time periods) is a costly process, and performed just once as an off-line process and then only the statistics have to be retrieved at query time.

For each time partition p_j in T_{LM}, the similarity score between q and p_j is computed (line 7). The similarity score is calculated using a normalized log-likelihood ratio (Equation 3.5). Each time partition p_j and its computed score are stored in C, or the set of time intervals and scores (line 8). After computing the scores for all time partitions, the contents of C will be sorted by similarity score, and then the *top-m* time intervals are selected as the output set A (line 10).

Algorithm 1 *DateQueryKeywords*($q, g, m, \mathcal{D}_{\mathcal{N}}$)

```
INPUT: Query $q$, time granularity $g$, number of time intervals $m$, and
    temporal corpus $\mathcal{D}_{\mathcal{N}}$
OUTPUT: Set of time intervals associated to $q$
$A \leftarrow \emptyset$                                  // Set of time intervals
$C \leftarrow \emptyset$                        // Set of time intervals and scores
$T_{LM} \leftarrow BuildTemporalLM(g, \mathcal{D}_{\mathcal{N}})$
for each $\{p_j \in T_{LM}\}$ do
    $score_{p_j} \leftarrow CalSimScore(q, p_j)$ // Compute similarity score of $q$ and $p_j$
    $C \leftarrow C \cup \{(p_j, score_{p_j})\}$          // Store $p_j$ and its similarity score
end for
$A \leftarrow C.selectTopMIntervals(m)$ // Select top-$m$ intervals ranked by scores

return $A$
```

Finally, the determined time intervals resulting from Algorithm 1 are assigned weights indicating their importance. A simple weighting method is computed, e.g., by assigning its reverse ranked number to each time interval.

The second approach to query dating can be performed in a similar manner, that is, instead of dating the query q directly, they use the *top-k* retrieved documents of the query q for dating a query. The resulting time of the query is the combination of determined times of each top-k document. The last approach is a variant of the dating using *top-k* documents described above. The idea is similar in the use of the *top-k* retrieved documents of the query q. The resulting time of the query is the creation date (or timestamps) of each top-k document. In this case, no temporal language models are used. The results from dating queries are used for re-ranking documents in favor of the determined time to improve the overall retrieval effectiveness.

An alternative to determine temporal query intent is to estimate the temporal part of a query by extracting temporal information within the document's content. This includes looking for temporal information within document content. Unlike metadata-based approaches, e.g., (Jones and Diaz, 2007; Kanhabua and Nørvåg, 2010a), the content-based approach (Campos et al., 2012a; Kanhabua et al., 2012; Strötgen et al., 2012) implies an increased level of difficulty since it usually involves linguistic analysis of texts. The simple identification of year patterns as done in (Metzler et al., 2009; Zhang et al., 2010) may not be enough to determine whether a date is relevant to the query because of the sparsity of such temporal patterns in query logs.

A key technique to overcome this problem is the extraction of temporal expressions using a time and event recognition algorithm, e.g., (Strötgen and Gertz, 2010; Verhagen et al., 2005). The algorithm extracts temporal expressions mentioned in a document and normalizes them to dates so they can be anchored on a timeline. Nevertheless, most state-of-the-art methodologies rely on existing temporal annotation tools, considering any occurrence of temporal expressions in web documents and other web data, as equally relevant to an implicit temporal query. For this reason, recent works have studied how to determine the relevance of temporal expressions for a given topic (Campos et al., 2012a; Kanhabua et al., 2012; Strötgen et al., 2012). The task of identifying relevant time can be regarded as a *classification problem*, i.e., to determine whether a temporal expression is *relevant* or *irrelevant* with respect to a query or an event. More precisely, the work of Campos et al. (2012a) can determine time for any type of query, while the work of Kanhabua et al. (2012) is specifically devoted to real-world events.

Campos et al. (2012a) identify top relevant *years* for a given query by analyzing top-k retrieved web snippets. The intuition is that web snippets are an interesting alternative collection for the representation of web documents, providing a short summary of the document, where dates, especially in the form of years, often appear. In particular, they propose a two-step approach: (1) generic temporal evaluation measure (GTE) evaluates the temporal similarity between a query and a candidate date, and (2) a classification model (GTE-Class) accurately relates relevant dates to their corresponding query terms and filter out non-relevant ones. In this regard, two different solutions were presented: a threshold-based classification strategy and a supervised classifier based on a combination of multiple similarity measures. Furthermore, Campos et al. (2014a) introduce a search interface that displays document results in a timeline, clustered using temporal expressions extracted from their corresponding snippets.

Kanhabua et al. (2012) employ a machine learning method for learning the relevance of temporal expressions using three classes of features: sentence-based, document-based and corpus-specific features. We review their approach in more detail as follows. Given a temporal expression the values of features are determined from the sentence s_i containing t_e, where s_i is extracted from an annotated document $\hat{d}$. The intuition is to determine the relevance of temporal expressions by considering *the degree of relevance of their corresponding sentences* with respect to a given event. For example, a sentence that is too long or too short is likely to be less relevant, and a sentence containing too many of place names or locations (e.g., cities, provinces and states) is possibly irrelevant or less specific to an event. The granularity type of time mentioned in a sentence can also indicate the relevance to a particular event, e.g., a time point should be more relevant than a time period because it is more precise/accurate. The features that can be extracted from the document level aimed at capturing the *ambiguity* of a document mentioning about a given event. These features can be computed off-line because they are independent from an event of interest.

Finally, non-relevant temporal information can be filtered out using some heuristics determined with respect to a particular document collection, denoted *corpus-specific*. In a domain-specific application like *early detection of real-world outbreak events* (Kanhabua and Nejdl, 2013), temporal expressions mentioned in question or negative sentences as well as those related to commercial or vaccinate campaigns can be ignored or regarded as *non-relevant* since they are not useful for the task. For that purpose, a list of *negative keywords* corresponding to irrelevant aspects is manually built and considered as a feature for classifying a given temporal expression. In addition, temporal expressions are not related to a given event if they refer to a past event, i.e., not referring to the current event of interest. Thus, they consider any term related to historical data (e.g., “statistic”, “annually”, “past year”) to be non-relevant to the query. For more detailed information about the proposed feature, please refer to the original work by Kanhabua et al. (2012).

4.2 Dynamic Query Subtopics

The dynamic of query subtopics and its impact on search has been studied in some recent works (Nguyen and Kanhabua, 2014; Whiting et al., 2013; Zhou et al., 2013). In order to illustrate the dynamics of query subtopics, we provide an example of the query ncaa. By analyzing query logs, we can observe on the early stage some queries related to the *pre-event* aspects of the event, e.g., *2006 ncaa tournament bracket*, and *march madness schedule*. Over time, we speculate change in query popularity of a new subtopic like *ncaa women's basketball tournament bracket*, which reflects an ongoing event. Finally, the subtopics with underlying post-event aspects, such as, *ncaa basketball results* and *ncaa final four* for *late-event* are anticipated in the later stage of the event.

Whiting et al. (2013) point out that event-driven topics contain highly variable subtopics, which make it difficult to determine the underlying temporal search intent. They propose an approach to identify all possible query subtopics from structured data represented by the section hierarchy of Wikipedia articles. They further investigate whether temporal changes in a section's content reflect the temporal query popularity (as also seen in query logs).

Zhou et al. (2013) study the effects of temporal subtopics on the evaluation of search result diversification. They argue that current diversity evaluation measures take the popularity of the subtopics into account and aim to favor systems that promote most popular subtopics earliest in the result ranking and subtopic popularity is assumed to be static over time. In fact, the temporal subtopic dynamics impact the traditional diversity metrics, especially for topically ambiguous temporal queries. In order that, this temporal characteristic should be considered when evaluating diversity. Additionally, they conduct a small study on the Wikipedia disambiguation pages to analyze changes in a subtopic popularity by analyzing the number of page views over time.

Nguyen and Kanhabua (2014) mine dynamic subtopics from two sources (query logs and a document collection). Then, they leverage the subtopics for time-aware search result diversification aimed at returning a diverse list of documents dynamically. Next, we describe their work, which is the current state-of-the-art on this topic in more detail.

4.2.1 Mining Subtopics from Historical Query Logs

Nguyen and Kanhabua (2014) make use of a set of queries Q, a set of URLs D and click-through information S. Each query $q \in Q$ is made up of query terms issued at hitting time q_t and associated with a set of subtopics C. A clicked URL $u_q \in D$ refers to a web document returned as an answer for a given query q, which has been clicked by the user. The click-through information composes of a query q, a clicked URL u_q, the position on result page, and its timestamps.

A bipartite graph is constructed using the history information with regard to different time points. Each subtopic is assigned a *temporal weight* that reflects the probability of the relevance of a subtopic at particular time, i.e., a weighted directed bipartite graph $G = (Q \cup D, E)$, where edges E represent the click-through information from a set of queries Q to a set of URLs D. Edges are weighted using click frequency - inverse query frequency (CF-IQF) model (Deng et al., 2009). CF-IQF compensates for common clicks on less frequent but distinguished URLs over common clicks on frequent URLs. Random walk with restart on the click-through graph provides a set of related queries, which are further processed by a clustering technique in order to remove duplicates.

To achieve finer-grained query subtopics at hitting time q_t, the acquired queries are clustered in a similar manner as performed in (Song et al., 2011). The steps are as follows: (1) construct a query similarity matrix (using lexical, click-based and semantic similarity), (2) cluster related queries using affinity propagation (AP), and (3) use the centroid of a cluster as a subtopic.

For a given temporal query q, the weight of subtopic $\hat{c}$, which is a representative of the cluster $\mathbb{C}$, can be calculated as the proportion between the sum of relatedness scores of all queries in $\mathbb{C}$, and that of all queries from all subtopic clusters.

$$w(\hat{c}) = \frac{\sum\limits_{c \in \mathbb{C}} relatedness(c, q)}{\sum\limits_{i=1}^{k} \sum\limits_{c \in \mathbb{C}_i} relatedness(c, q)}, \tag{4.12}$$

Table 4.2: Top-10 dynamic query subtopics ordered by weights for the query ncaa.

March 14		March 31		April 07	
subtopic c	$w(c)$	subtopic c	$w(c)$	subtopic c	$w(c)$
march madness schedule	0.0132	oakland raiders	0.0100	*ncaa women's basketball tournament*	0.0122
ncaa basketball tournament	0.0117	ncaaw	0.0090	ncaa basketball tournament	0.0053
nfl draft	0.0068	tito francona	0.0042	cbs sports line	0.0049
selection sunday	0.0048	ncaa brackets	0.0031	*ncaaw*	0.0033
oakland raiders	0.0037	ncaa division ii	0.0029	*ncaa final four*	0.0031
2006 ncaa tournament bracket	0.0032	andy goram	0.0024	ncaa wrestling	0.0029
brad hopkins released nfl	0.0026	lakers	0.0024	march madness bracket	0.0028
roger clemens	0.0023	*ncaa women's basketball tournament bracket*	0.0024	*ncaa basketball results*	0.0019
ncaa division ii	0.0021	ncaa basketball brackets	0.0021	andy goram	0.0009
college basketball	0.0014	nit brackets	0.0021	ncaa division ii	0.0009

where *relatedness*(c,q) is measured as a RWR score of c with respect to q. The parameter k is the number of clusters determined by AP clustering algorithm.

Table 4.2 shows the temporal subtopics of the query ncaa mined from the AOL query log at three different hitting times. Subtopics retrieved on March 14, 2006 reflect *pre-event* aspects, such as, *march madness*, whereas the subtopic *ncaa women's basketball tournament bracket* appears in the top-ranked list of March 31 since it refers to an ongoing event began around March 18. Interestingly, various aspects are *post-event* or refer to the later stage of the tournament have emerged on April 7, namely, *ncaa basketball results* and *ncaa final four*.

4.2.2 Mining Subtopics from a Document Collection

An alternative method is mining dynamic subtopics from a temporal document collection. Nguyen and Kanhabua (2014) make use of Latent Dirichlet Allocation (LDA) (Blei et al., 2003), an unsupervised method to model latent query subtopics from a set of relevant documents D at a particular time period. Each subtopic $c \in C$ is modeled as multinomial distribution of words, a document $d \in D$ composes of a mixture of topics. There are several reasons for choosing LDA for mining latent subtopics. The first advantage of using LDA on a query-based set of documents is that subtopic directly mined from the collection have less noise, better quality and reflect the *intrinsic* aspects underlying a target collection. The second reason is that LDA is a probabilistic-based model, and its output can directly be incorporated into probabilistic subtopic modeling, as followed the approach proposed by (Carterette and Chandar, 2009).

Deciding the optimum number of subtopics is an important issue and the number is subject to change at different time periods, which reflects multiple, dynamic aspects associated to a query. LDA allows to model the topic distribution of a given document collection, but the number of topics is a fixed parameter. However the number of aspects underlying a given query is often not known in advance. There have been several works (Cao et al., 2009; Arun et al., 2010) on finding the optimal number of topics for LDA. Nguyen and Kanhabua (2014) follow the approach that proposed by Arun et al. (2010) to identify the number of latent subtopics that are naturally present in each partition. They view LDA as a matrix factorization mechanism. Given a document collection C, it is split into two matrix factors $M1$ and $M2$ as given by $C_{\mathbf{d}\times\mathbf{w}} = M1_{\mathbf{d}\times\mathbf{t}} \times M2_{\mathbf{t}\times\mathbf{w}}$, where $\mathbf{d}$ is the number of documents present in the corpus and $\mathbf{w}$ is the size of the vocabulary. The quality of splits depends on $\mathbf{k}$, i.e., the right number of topics chosen. The measure is computed in terms of symmetric KL-Divergence of salient distributions that are derived from these matrix factors. They observed that a non-optimal number of topics produces a high divergence value.

$$Divergence(M1, M2) = KL(C_{M1}|C_{M2}) + KL(C_{M2}|C_{M1}), \tag{4.13}$$

where C_{M1} is the distribution of singular values of topic-word matrix M1. C_{M2} is the distribution obtained by normalizing the vector $L \cdot M2$, where L is $1 \times D$ vector of lengths of each document in the corpus and $M2$ is the document-topic matrix. A non-optimum number of subtopics produces high divergence between the salient distributions derived from two matrix factors, i.e., topic-word and document-topic. The number of topics is set in a pre-defined value range from γ to δ, and the optimal number of topic has the minimum KL-divergence value.

Table 4.3: Subtopics associated to the query *apple* using LDA-based subtopic mining from TREC Blog08.

IPHONE		MACBOOK		FRUIT		PIE	
word w	$P(w\mid c)$	word w	$P(w\mid c)$	word w	$P(w\mid c)$	word w	$P(w\mid c)$
iphone	0.652	macbook	0.999	apple	0.397	apple	0.480
software	0.276	apple	0.511	tree	0.192	pie	0.220
developers	0.220	pro	0.404	trees	0.118	butter	0.185
app	0.212	nvidia	0.280	fruit	0.110	recipe	0.181
nda	0.192	graphics	0.252	garden	0.092	sugar	0.134
store	0.143	air	0.137	varieties	0.056	juice	0.125
application	0.120	ghz	0.127	growing	0.045	cup	0.116

The weight of subtopic c can be estimated at every hitting time. The weight $w(c)$ reflects the probability that a given query q implies c. The probability that q belongs to c, $P(c|q)$, is determined as the popularity of the subtopic in the studied time slice of the document collection. It is calculated as the proportion between the total probabilities of all documents belongs to a subtopic, and the number of documents in the time slice.

$$P(c|q) = \frac{\sum\limits_{d \in D_t} P(c|d)}{|D_t|}, \tag{4.14}$$

where $P(c|d)$ is calculated from the Dirichlet prior topic distribution of LDA as follow.

$$P(c_i|d_j) \propto P(d_j|c_i)P(c_i) = \prod P(w|c_i)\ P(c_i), \tag{4.15}$$

where $P(w|c_j)$ and $p(c_j)$ can be determined using Gibbs sampling proposed by Griffiths and Steyvers (2004). The example output of subtopics mining using LDA for the query apple is illustrated in Table 4.3. There are four of the retrieved subtopics presented, namely, *iphone*, *macbook*, *fruit* and *pie*. Each subtopic is represented by the list of words ranked by its probability of belonging to the subtopic.

4.3 Time-aware Query Enhancement

Query enhancement refers to IR techniques for improving the overall retrieval effectiveness or increasing a positive user experience. Basic methods for query enhancement techniques are, for example, pseudo-relevance feedback, query expansion, and query reformulation (Baeza-Yates and Ribeiro-Neto, 2011; Croft et al., 2009; Manning et al., 2008). In this section, we review current query enhancement studies in the temporal information retrieval paradigm, which include temporal relevance feedback, temporal query performance prediction, as well as time-aware query reformulation.

Temporal Relevance Feedback

Retrieval effectiveness can be increased by employing *pseudo-relevance feedback* (PRF), which can be done in two steps. First, the initial search is performed for a given query, where a set of top-k retrieved documents are assumed to be relevant. Second, terms are extracted from those top-k documents and the query is automatically expanded with extracted terms for performing a second search that delivers the final results. When taking the time dimension into account, PRF is known as *time-based pseudo-relevance feedback* (Amodeo et al., 2011a; Dakka et al., 2012; Keikha et al., 2011a,b; Peetz et al., 2014).

Keikha et al. (2011a,b) propose a time-based query expansion technique that selects terms for expansion from different times. Then, the technique was used for retrieving and ranking blogs, which also captures the dynamics of the topic both in aspects and vocabulary usage over time. Amodeo et al. (2011a) select top-ranked documents in the highest peaks as pseudo-relevant, while documents outside peaks are considered to be non-relevant. They use Rocchio's algorithm for relevance feedback based on the top-10 documents.

Dakka et al. (2012) incorporate temporal distributions in different language modeling frameworks. While they do not actually detect events, they perform several standard normalizations to the temporal distributions, i.e., using global temporal distributions as a prior. Peetz et al. (2014) consider temporal bursts extracted from top-k search results for improving query modeling. Particularly, they assume that documents occurring within bursts more likely to be relevant than those outside of bursts, namely, documents within bursts contribute more useful terms for query modeling than documents selected for relevance models. They identify temporal bursts in ranked lists of initially retrieved documents for a given query and model the generative probability of a document given a burst. They propose various discrete and continuous models and sampled terms from the documents in the burst and update the query model. The effectiveness of their query modeling approaches is assessed using several test collections [2] based on news articles (TREC-2, 7, and 8) and a test collection based on blog posts (TREC Blog track, 2006-2008), both for time-aware queries and for arbitrary queries. For query sets that consist of both temporal and non-temporal queries, the proposed temporal burst query modeling is able to find the balance between performing query modeling or not: only if there are bursts and only if some of the top ranked documents are in the burst, the query is remodeled based on the bursts.

In the context of real-time search, Efron et al. (2014) present the temporal cluster hypothesis for tweet search. In their study, they retrieve initial search results using a baseline retrieval model, and determine the temporal density of relevant documents for re-ranking search results. Through experiments on TREC datasets, temporal feedback based on the temporal density function using kernel density estimation improves search effectiveness over lexical (i.e., content-based) approaches. Moreover, they also point out a direction where a further investigation can be done, namely, documents with similar "temporal profiles" are likely to relevant to the same information need; thus providing temporal signals for more effective ranking.

Temporal Query Performance Prediction

Automatically applying a query enhancement technique can lead to query drift and possibly lower retrieval effectiveness. For that reason, a search system should be able to make a decision whether a particular enhancement technique can be taken to improve the overall performance, or choose between alternative query enhancement techniques, such as, query expansion and query suggestion. An automatic method for such decision-making is called *query performance prediction*, i.e., predicting

[2] `http://trec.nist.gov`

the retrieval effectiveness that temporal queries will achieve with respect to a particular ranking model in advance of, or during the retrieval stage in order that particular actions can be taken to improve the overall performance (Hauff et al., 2010; Carmel and Yom-Tov, 2010).

Different approaches to predicting query performance can be categorized according to two aspects Hauff et al. (2008): 1) time of predicting (pre/post-retrieval) and 2) an objective of task (difficulty, query rank, effectiveness). Pre-retrieval based approaches predict query performance independently from a ranking method and the ranked list of results. Typically, pre-retrieval based methods are preferred to post-retrieval based methods because they are based solely on query terms, the collection statistics and possibly external sources, e.g., WordNet or Wikipedia. On the contrary, post-retrieval based approaches are dependent on the ranked list of results. Pre-retrieval predictors can be classified into four different categories based on the predictor taxonomy defined by Hauff et al. Hauff et al. (2009): 1) specificity, 2) ambiguity, 3) ranking sensitivity, and 4) term relatedness.

Formally, the task of *temporal query performance prediction* can be defined as follows. Let q be a temporal query, D be a document collection, T be a set of all temporal expressions in D. N_D is the total number of documents in D and N_T is the number of all distinct temporal expressions in T. Temporal query performance prediction is aimed at predicting the retrieval effectiveness for q. Because q is strongly time-dependent, both the statistics of the document collection D and the set of temporal expressions T must be taken into account. Temporal query performance prediction is defined as $f(q, D, T) \rightarrow [0, 1]$, where f is a prediction function (so-called a predictor) giving a predicted score that can indicate the effectiveness of q. Ultimately, it is aimed at finding f that can best predict the effectiveness of q, i.e., predicted scores are *highly correlated* with actual effectiveness scores.

In general, f can be learned using simple linear regression, which models the relationship between the effectiveness y with a single predictor variable p. Given N queries, simple linear regression fits a straight line through the set of N points of effectiveness scores versus predicted scores. So that, the sum of squared residuals of the model (or vertical distances between the points of the data set and the fitted line) is as small as possible.

Jones and Diaz (2007) employ temporal features extracted from top-k retrieved documents to predict the mean average precision of a query. The experimental results showed that adding temporal features to a regression increases the predictive performance. Similarly, Kanhabua and Nørvåg (2011) propose different time-based predictors and combine the proposed predictors for improving the predictive performance using linear regression and neural networks. In their work, the studied features for predicting query performance belong to the pre-retrieval class, which can be determined independently from a ranking method as opposed to post-retrieval predictors.

In line with the previous work on time-aware query performance prediction, Tran et al. (2015) investigate novel features for prediction that explicitly take the temporal dimension into account, e.g., temporal document frequency, temporal scope and temporal similarity. Their proposed method is different from the previously mentioned approaches focusing on precision metrics, while they consider the performances of queries in terms of recall, which have been recently remarked and considered in different information retrieval scenarios (Cormack and Grossman, 2014; Li et al., 2014).

Time-aware Query Reformulation

When searching a temporal document collection, search results can be affected by terminology evolution (Tahmasebi et al., 2008). Precisely, the evolution includes the changes of words related to their definitions, semantics, and names (people, location, etc.). This problem can be regarded as query expansion and query reformation for temporal queries. It is important to note that language changing is a continuous process that can be observable also in a short term period.

The variation in languages causes two problems in text retrieval; 1) *spelling variation* or a difference in spelling between the modern and historic language, and 2) *semantics variation* or terminology evolution over time (new words are introduced, others disappears, or the meaning of words changes). In order to search historic texts using contemporary language, a common approach is to use query expansion employing name variant lists mapping current to historic terms. These lists can either be generated manually (Rayson et al., 2005) or automatically. Manual approaches are very labor-intensive, and are in general most useful for languages where institutions have defining spelling standards (as has been the case in, e.g., Spain and France), since that will limit the number of name variants to be found. For languages which only in modern times have had standardized spelling (e.g., German and English) automatic approaches are preferred.

Ernst-Gerlach and Fuhr (2006, 2007) developed an approach for semi-automatically generating search term variants for text collections with historic spellings. Generation of the name variants is performed using a set of transformation rules. The starting point for creating the rules is a training sample of historic texts. A spell checker for contemporary German is used to detect terms that were spelled different from today, and a mapping from old to contemporary variant is created manually. Finally, transformation rules are created based on these mappings. A simple example is a mapping *(unnütz, unnuts)* that can contribute to a rule *ü* → *u*. While still having a significant manual part, new texts added to a collection will require less manual work since many of the terms not already known can be handled automatically based on the existing rules. The approach has only been applied on German texts so the accuracy when employed on other languages is unknown. Also, the approach is not able to detect homographs (ancient spelling that matches a different contemporary word).

The approach of Ernst-Gerlach and Fuhr is well designed to handle terms that changes gradually over time. However, some terms can change their spelling completely and independent of any rules that can be created. This change can often be seen for city names (e.g., *Leningrad* and *Saint Petersburg*) and company names, and words being borrowed from other languages. Two approaches for determining mappings of such terms are the use of co-occurrence statistics and using information implicitly stored in knowledge bases.

The study of semantic changes over time has been addressed extensively in previous works. Berberich et al. (2009) propose a method based on a hidden Markov model for reformulating a query into terminology prevalent in the past. More precisely, they exploit co-occurrence statistics in order to determine *across-time semantic similarity*. For example, the fact that the terms *portable, music, earphones* appear frequently together with *Walkman* in 1990 and with *iPod* in 2005 gives them high across-time semantic similarity, so that if a query contains iPod or Walkman it should be expanded with the other term. Kaluarachchi et al. (2010a,b) study the problem of concepts (or entities) whose names can change over time. They propose to discover concepts that evolve over time using association rule mining, and used the discovered concepts to translate time-sensitive queries and answered appropriately. de Boer et al. (2010) present a method for automatically extracting event time periods related to concepts from web documents. In their approach, event time periods are extracted from different documents using regular expressions, such as, numerical notations for years.

Name variant lists can be generated from information implicitly stored in knowledge bases. For example, Bøhn and Nørvåg (2010) present a three-step approach that uses contents and links in Wikipedia. In a first step they extract named entities. Second, they use internal links redirects and disambiguation pages to find candidates for name variants. Finally, a filtering step removes those candidates considered "noise".

While name variant lists are useful for time-focused search and ranking purposes it is also desired that name variants should have attached temporal intervals, representing the time period that name variant was valid. An approach for generating time-dependent name variant lists automatically was first developed in (Kanhabua and Nørvåg, 2010b,c). Their approach takes advantage of the availability of the complete Wikipedia history and the possibility to create snapshots of Wikipedia as it was at a certain point in time. In the first step, they create all snapshots for a certain granularity, for example each month. Second, for all snapshots they create name variant lists using a variant of the one presented in (Bøhn and Nørvåg, 2010). Then, the time period for each name variant is determined based on the Wikipedia snapshots where the name variant existed. However, the time periods of entitysynonym relationships do not always have the desired accuracy. The main reason for this is that the Wikipedia history has a relatively short time span. As a consequence, Kanhabua and Nørvåg (2010b) improve the accuracy of the initial time periods by performing burst detection on a temporal document collection covering a longer time period, such as, the New York Times Annotated Corpus. The detected burst periods are used to represent periods that the name variants are in use over time. The described approach and also be used to distinguish between time-dependent and time-independent name variants (i.e., whether the association changes over time or not).

The main limitation of the approach by Kanhabua and Nørvåg (2010b) is that an external knowledge source like Wikipedia does not cover all entities and are not able to capture ephemeral names or jargon used in everyday language or social media. Tahmasebi et al. (2012) propose to automatically detect terminology evolution within large, historic document collections by using clustering techniques and analyzing co-occurrence graph. This work was later extended with a filtering method that makes use of semantic web resources in order to better handle more informal texts like blogs, which typically are closer to spoken language (Holzmann et al., 2013). A different approach for searching historical texts using contemporary language, is to consider it as a variant of cross-language retrieval. Efron (2013) approaches this task by proposing several techniques, the best-performing one is a feed-back technique that combines several types of evidence, like a bilingual dictionary containing a list of mapping from archaic to modern English words. It is uncertain, however, to what extent the approach can be applied to highly inflected languages.

There are more recent works on temporal analytics of words. For example, Jatowt and Tanaka (2012) study the evolution of English language vocabulary over the last two centuries to help with understanding its impact on readability and retrieval of historical documents. Radinsky et al. (2011) leverage a news archive for computing the semantic relatedness of words, and they propose a new method based on time-series analysis, so-called Temporal Semantic Analysis (TSA), which explicitly models temporal information in order to measure semantic relatedness.

4.4 Summary

We have presented a taxonomy of time-sensitive queries and illustrate some examples for different types. We provided a systematic review of previous works on temporal query analysis and outline their underlying methodologies in detail. The problems and tasks addressed in the previous works are varied, e.g., query intent understanding, detection, classification or prediction. Determining user search intent is an important task for providing a better search experience, i.e., the system should understand latent search intent.

Existing methods for predicting user search intent relied heavily on analyzing the historical query logs. However, a limitation of using query logs is the sparsity of data. Thus, another direction is to conduct time-series analysis on document collections in order to overcome the shortage of using query logs only. We classify state-of-the-art on determining temporal query intent into two main categorizes based on the data sources used, namely, 1) mining temporal patterns in query streams, and 2) determining temporal intent from top-k search results.

In some special cases, a temporal query contains dynamic subtopics, so-called temporally ambiguous. For instance, the query US Open is more likely to be targeting the tennis open in September and the golf tournament in June. More precisely, users' search intent can be identified by the popularity of a subtopic with respect to the time where the query is issued. We give an overview of recent works on analyzing the temporal variability of query subtopics by applying subtopic mining techniques at different time periods. The analysis results reveal that the popularity of query aspects changes over time, which is possibly the influence of a real-world event.

Understanding the temporal search intent behind a user's query is the first step before applying query enhancement techniques and selecting a suitable time-aware retrieval and ranking model. Query enhancement refers to IR techniques aimed at improving the overall retrieval effectiveness or increasing search user experiences. To this end, we have presented enhancement techniques including temporal relevance feedback, time-aware query performance prediction, as well as time-aware query reformulation.

5

Time-aware Retrieval and Ranking

The search results over a temporal collection might contain a large amount of irrelevant information. In order to support a temporal search, a basic solution is to extend keyword search with the creation or published date of documents (Berberich et al., 2007; Nørvåg, 2004). In that way, search results are restricted to documents from a particular time period given by a time constraint, ranked by textual relevance and displayed in chronological order with recently created documents ranked higher than older documents. Nevertheless, chronological ordering is not always effective because a user must spend a non negligible amount of time exploring retrieved results in order to find those satisfying her information need. In order to increase the retrieval effectiveness, the time dimension should be explicitly incorporated into retrieval and ranking models, so-called *time-aware retrieval and ranking.* More precisely, documents must be ranked according to both textual and temporal similarity with respect to given temporal information needs.

Existing works on time-aware ranking can be classified into different types based on two main notions of relevance with respect to time: 1) recency-based ranking, and 2) time-dependent ranking. Recency-based ranking methods promote documents that are recently created or updated. The preference of freshness is quite common in a general web search, where a user looks for recent information, e.g., about breaking news. On the other hand, time-dependent ranking methods takes into account the relevant time periods underlying a given temporal query, which can be determined using the temporal query analysis methods presented in Section 4. This type of ranking explicitly adjusts the score of a document with respect to the time of queries, for example, by assuming that documents with creation dates close to the query's time are more relevant and thus must be ranked higher.

In addition to the aforementioned classification of time-aware methods, we also outline existing works that explicitly model entities and events into retrieval and ranking processes. Note that, entity and event-based retrieval returns events instead of documents, whereas time-aware ranking methods use a very simple notion of events.

5.1 Recency-based Ranking

Li and Croft (2003) propose to incorporate time criteria into a language modeling framework (Lavrenko and Croft, 2001; Ponte and Croft, 1998). In the aforementioned language modeling approaches for ranking, it is assumed uniform document prior probabilities, but in the temporal language model, they assign prior probabilities with an exponential function of the created date of a document where a document with a more recent creation date obtains high probability. In order to evaluate their proposed method, they manually select 36 recency queries (see Appendix A for the description of recency queries) from topics 301-400 over TREC-7,8. In this work, they did not explicitly use the contents of documents, but only date metadata.

In the following, we outline language model approaches that incorporate a temporal document prior. Li and Croft (2003) propose to replace $p(d)$ in Equation 3.2 with some probability dependent on document's publication date, i.e., $p(d|t_d)$. This gives us the time-based language models:

$$p(d|t_d) = DecayRate^{\lambda \cdot \frac{|t_C - t_d|}{\mu}}, \tag{5.1}$$

where $DecayRate$ and λ are constants; $0 < DecayRate < 1$ and $\lambda > 0$. μ is a unit of time distance. t_C is the most recent date (in months) in the whole collection, $t_d = PubTime(d)$. On the other hand, Peetz and de Rijke (2013) consider different temporal document priors inspired by retention functions (Meeter et al., 2005) considered in cognitive psychology that are used to model different *forgetting* functions. They conducted the experiments using 20 recency queries from topics 101-200 of TREC-2, 16 recency queries from topics 301-350 of TREC-6, and 24 recency queries from topics 351-450 of TREC-7,8. Their results show that the Weibull distribution-based function outperforms other decay functions. The Weibull function can be computed as:

$$f_{extended\ Weibull}(d, q, g) = b + (1 - b)\mu e^{-\frac{a\delta g(d,q)^s}{s}}, \tag{5.2}$$

where a and d indicate how long the item is being remembered: a indicates the overall volume of what can potentially be remembered, s determines the steepness of the forgetting function; μ determines the likelihood of initially storing an item, and b denotes an asymptotic parameter. Note that, we use the notation s to represent the steepness parameter, whereas d is used in (Peetz and de Rijke, 2013). This retention function is used to compute a temporal document prior $p(d|t_d)$, which will be integrated into a retrieval model. The recommended values of parameters (optimized for TREC-6) are: $a = 0.009$, $s = 0.7$, $b = 0.1$, and $\mu = 0.7$.

There are previous works that improve link-based web authority methods by incorporating freshness information. Previous works on time-aware ranking that exploits link structures are introduced in (Berberich et al., 2005; Corso et al., 2005; Dai and Davison, 2010a; Yu et al., 2004). Yu et al. (2004) point out that traditional link-based algorithms (i.e., PageRank and HITS) simply ignore the temporal dimension in ranking. They modify the PageRank algorithm by taking into account the date of a citation in order to improve the quality of publication search. A publication obtains a ranking score by accumulating the weights of its citations, where each citation receives a weight exponentially decreased by its age. Corso et al. (2005) propose a ranking algorithm of news stream, finding the most authoritative news sources and identifying the most interesting events in the different categories.

Based on the fact that a news article can be aggregated to the news article previously posted, a virtual linking relationship between pieces of news and news sources can be analyzed from the news posting process or the natural aggregation by topics between different new stories. They assume the importance of new article changes over time based on its freshness. For this purpose, they represent news creation as an undirected graph where nodes are news articles and their news source using two kinds of edge: 1) connecting news sources and articles, 2) connecting similar pieces of news after a clustering process. The ranking scheme depends on two parameters, p is the decay rate of freshness of news article, and b is the amount of a source's rank we want to transfer to each posted piece of news. By studying sensitivity of the ranks obtained by varying two parameters, it shows that the algorithm is robust as the correlation between ranks remains high with respect to content changes.

Berberich et al. (2005) also extend PageRank to rank documents with respect to freshness. The difference is that this work defines freshness as a linear function that will give a maximum score when the date of document or link occur within *the user specified period* and decrease a score linearly if it occurs outside the interval. In more recent work, Dai and Davison (2010a) study the dynamics of web documents and links that can affect relevance ranking, and proposed a link-based ranking method incorporating the freshness of web documents. Intuitively, features used for ranking are captured by considering two temporal aspects: 1) how fresh the page content is, referred to as page freshness, and 2) how much other pages care about the target page, referred as in-link freshness.

Another recency-based method quantifies the freshness of documents by analyzing temporal content dynamics and incorporates freshness into ranking. Jatowt et al. (2005) present an approach to rank a document by its freshness and relevance. The method analyzes content that has changed between a current version and archived versions, assigning a similarity score of changes to a query topic. It is assumed that a document is likely to have fresh contents if it is frequently changed and on-topic. Thus, documents are ranked with respect to the relevance of changed contents to the topic, the size of changes and the time difference between consecutive changes. In other words, a document is ranked high if it is modified significantly and recently. Aji et al. (2010) introduce a new term weighting model that uses the revision history analysis (RHA) of a document to redefine a term's importance, assuming that a term should be as relevant as the number of times it occurs in the different versions of a document gets higher. They include a decay factor so that the terms in older versions of the document get a higher value. RHA is then incorporated into BM25 and statistical language models and documents get ranked based on the importance of the terms in the past. Elsas and Dumais (2010) study the relation between temporal dynamics of document's contents and relevance ranking. They proposed a language model that takes into account the changing document content. In the language model, terms are weighted differently based on their temporal characteristic. They also propose a query independent document prior that favors dynamic documents. The two proposed systems give a significant improvement on the *navigational queries*, which are considered dynamic content-prone.

There are several previous works employing feature-based methods and machine learning (Dai et al., 2011; Dong et al., 2010a,b; Whiting and Jose, 2014; Shokouhi and Radinsky, 2012). Dong et al. (2010a) propose a system that automatically detect and classify recency-sensitive queries like queries about breaking news stories. Recency-sensitive queries detection used three types of recency features: *timestamp features*, *linktime features* (extracted from link-based page discovery log that provides page freshness evidence) and *WebBuzz features* (detect recency URLs). The training data is then used to learn a ranking function by combining multiple temporal features. On the contrary, Dai et al. (2011) propose a framework where each query is run against a set of rankers. Consequently, weights vary based on the temporal profile of a query, thus minimizing the risk of poor performance when queries are misclassified in terms of recency intent.

Dong et al. (2010b) leverage features from micro-blog data, i.e., Twitter for the task of real-time recency ranking. The features are categorized into three different types: *textual features*, *social network* *features* (Twitter users network graph as adjacency matrix) and *other features* (based on the interaction between users and tweets). Styskin et al. (2011) propose to solve the recency ranking problem by using result diversification principles to deal with the queries which the need level of recent content is uncertain. They also propose a query classification model for recency sensitive queries. The model quantifies the recency need level of a particular query.

While most of previous works on recency ranking focus on real-world events or news queries, which typically exhibit bursts in the volume of published documents or submitted queries. Cheng et al. (2013), on the other hand, study the role of time in queries such as “credit card overdraft fees” that have no major spikes in either document or query volumes over time, yet they still favor more recently published documents (so-called *timely queries*). They show that the change in the terms distribution of results of timely queries over time is strongly correlated with the users’ perception of time sensitivity. Moreover, they propose a method to incorporate document freshness into a ranking model. Through experimentation, the proposed ranking model improves the retrieval effectiveness compared to other time-sensitive and non time-sensitive ranking algorithms for timely queries.

In addition to a document retrieval task, recency-based ranking also gains interest in query enhancement techniques, such as, time-aware query auto-completion.

Query auto-completion (QAC) is the task of suggesting queries based on the prior is the first part of the query as user types in. Recent works presented in (Shokouhi and Radinsky, 2012; Whiting and Jose, 2014) exploit the need of a *time-sensitive* query auto-completion that suggests query based on recency. One example is when user types in “k” in the search engine in September 2013. A time-aware query completion should take into account a recent peak in query volume for Kenya and rank it higher than a more common query like *Kim Kardashian*. Shokouhi and Radinsky (2012) propose to rank the query candidates based on their predicted popularity using a time-series technique, while Whiting and Jose (2014) tackle a similar problem using a machine-learning approach and their propose recency features.

In the following, we summarize *recency-based* features from recent works on query auto-completion.

Most popular completion is the most intuitive approach in QAC (Bar-Yossef and Kraus, 2011). The model can be regarded as an approximate Maximum Likelihood Estimator (MLE), that ranks the suggestions based on past query popularity. Let $P(q)$ be the probability that the query q will be issued byu a user. Given a prefix x, the query candidates that share the prefix $\mathcal{Q}_c$, the most likely suggestion $q \in \mathcal{Q}_c$ is calculated as: $MLE(x) = argmax_{q \in \mathcal{Q}_c} P(q)$

Recent MLE is proposed by Whiting and Jose (2014); Shokouhi and Radinsky (2012) does not take into account the whole past query log information like the original MLE, but uses only recent days, i.e., applying a sliding window. The popularity of query q in the last n days are aggregated to compute $P(q)$.

Last N query distribution considers the last N queries given a prefix x and time t. This approach was presented in (Whiting and Jose, 2014; Shokouhi and Radinsky, 2012) and it complements the weakness of recent MLE in a time-aware context while having to determine the size of the sliding window for prefixes with different popularities. In this approach, only the last N queries are used for ranking, of which N is the tuning parameter between *relevant* and *recency*.

Predicted next N query distribution employs the past query popularity as a prior for predicting query popularity at hitting time. The prediction is then applied for QAC as done in two previous works (Whiting and Jose, 2014; Shokouhi and Radinsky, 2012).

5.2 Time-dependent Ranking

This section provides an overview of time-dependent ranking methods. We then explain some selected approaches in more detail.

Perkiö et al. (2005) introduce a process of automatically detecting a topical trend (the strength of a topic over time) within a document corpus by analyzing temporal behavior of documents using a statistic topic model. Then, it is possible to use topical trends on top of any traditional ranking like TF-IDF to improve the effectiveness of retrieval.

Shaparenko et al. (2005) propose a method that does not require link information, which is appropriate for various types of documents, for example, emails or blogs, lacking meaningful citation data. The idea is to identify the most influent document, defining its impact as the amount of follow-up work it generates represented as *lead/lag index*. The index measures if a document is more of a leader or more of a follower by comparing similarities of two documents and time lag.

Metzler et al. (2009) mine query logs by analyzing query frequencies over time in order to identify strongly time-related queries. Moreover, they presented a ranking method concerning temporal information needs that are not provided by a query as such. A previous approach to exploiting the transient and bursty nature of relevance in temporally ordered document collections is presented in (Peetz et al., 2014). Similarly, Dakka et al. (2012) propose a method based on the analysis of temporally active time periods (salient events) in the temporal distribution of pseudo-relevant documents. Consequently, they leverage the determined temporal relevance for answering time-sensitive queries by seamlessly integrating into the BM25 ranking model. Keikha et al. (2011a,b) use relevance models of temporal distributions of posts in blog feeds. More precisely, they propose a time-based query expansion technique that selects terms for expansion from different times. Then, the technique was used for retrieving and ranking blogs, which also captures the dynamics of the topic both in aspects and vocabulary usage over time.

In the rest of this subsection, we present five time-dependent ranking methods, namely, LmT and LmtU from (Berberich et al., 2010), TS and TSU from (Kanhabua and Nørvåg, 2010a), and FuzzySet from (Kalczynski and Chou, 2005). We first explain some preliminaries for temporal ranking, and describe each method in more detail.

For a given temporal query, a time-dependent ranking method ranks documents that are textually and temporally similar to a query and ranks retrieved documents with respect to both similarities. Previous work has followed one of two main approaches: 1) a mixture model linearly combining textual similarity and temporal similarity, or 2) a probabilistic model generating a query from the textual and temporal part of a document independently. Temporal expressions and the publication date of a document is represented as a quadruple in (Berberich et al., 2010): (tb_l, tb_u, te_l, te_u), where tb_l and tb_u are the lower bound and upper bound for the begin boundary of the time interval respectively. Similarly, te_l and te_u are the lower bound and upper bound for the end boundary of the time interval respectively. A temporal query q is composed of keywords q_{text} and temporal expressions q_{time}. A document d consists of a textual part d_{text}, i.e., a bag of words, and a temporal part d_{time}, i.e., the publication date and temporal expressions $\{t_1, \ldots, t_k\}$.

In (Kanhabua and Nørvåg, 2010a), they propose a mixture model used to combine textual similarity and temporal similarity for ranking time-sensitive queries, which is also done in a similar manner in a more recent work by Campos et al. (2014c). Given a temporal query q with the determined time q_{time}, the score of a document d is computed as follows:

$$S(q, d) = (1 - \alpha) \cdot S'(q_{word}, d_{word}) + \alpha \cdot S''(q_{time}, d_{time}), \tag{5.3}$$

where the parameter α indicates the importance of textual similarity $S'(q_{word}, d_{word})$ and temporal similarity $S''(q_{time}, d_{time})$. Both similarity scores must be normalized using the maximum scores before combining them together to obtain the final score $S(q, d)$. $S'(q_{word}, d_{word})$ can be measured using any of existing text-based weighting functions. Similar to a query-likelihood approach, $S''(q_{time}, d_{time})$ measure temporal similarity by assuming that a temporal expression $t_q \in q_{time}$ is generated independently from each other, and a two-step generative model was used:

$$\begin{aligned} S''(q_{time}, d_{time}) &= \prod_{t_q \in q_{time}} P(t_q | d_{time}), \\ &= \prod_{t_q \in q_{time}} \left(\frac{1}{|d_{time}|} \sum_{t_d \in d_{time}} P(t_q | t_d) \right). \end{aligned} \tag{5.4}$$

Jelinek-Mercer smoothing will be applied to the above equation to avoid the zero-probability problem. $P(t_q|t_d)$ can be estimated using different temporal ranking methods.

The first method LmT ignores the time uncertainty, i.e., only temporal expressions exactly equal will be considered. $P(t_q|t_d)$ under LmT can be calculated as:

$$P(t_q|t_d)_{LMT} = \begin{cases} 0 & \text{if } t_q \neq t_d, \\ 1 & \text{if } t_q = t_d. \end{cases} \tag{5.5}$$

The method LmtU concerns the time uncertainty and assumes equal likelihood for each time interval t'_q that t_q can refer to. In other words, all time intervals $t_q = \left\{t'_q | t'_q \in t_q\right\}$ are assumed equally likely. The simplified calculation of $P(t_q|t_d)$ for LmtU is given as:

$$P(t_q|t_d)_{LMTU} = \frac{|t_q \cap t_d|}{|t_q| \cdot |t_d|}. \tag{5.6}$$

The detailed computation of $|t_q \cap t_d|$, $|t_q|$ and $|t_d|$ can be referred to (Berberich et al., 2010). The next temporal ranking method is TS, which also ignores the time uncertainty. The calculation of $P(t_q|t_d)_{TS}$ is actually equivalent to $P(t_q|t_d)_{LMTU}$. On the contrary, the method TSU takes uncertainty into account. When uncertainty is concerned, $P(t_q|t_d)_{TSU}$ is defined using an exponential decay function:

$$P(t_q|t_d)_{TSU} = DecayRate^{\lambda \cdot \frac{|t_q - t_d|}{\mu}}. \tag{5.7}$$

$$(t_q - t_d) = \frac{(tb_l^q - tb_l^d) + (tb_u^q - tb_u^d) + (te_l^q - te_l^d) + (te_u^q - te_u^d)}{4}. \tag{5.8}$$

where $DecayRate$ and λ are constants; $0 < DecayRate < 1$ and $\lambda > 0$. μ is a unit of time distance. The main idea is to give a score that decreases proportional to the difference between t_q and t_d. The shorter the distance between two temporal expressions or time points, the more temporally similar they are. The last method FuzzySet is proposed in (Kalczynski and Chou, 2005), and it measures temporal similarity using a fuzzy membership function. $P(t_q|t_d)_{FuzzySet}$ is given as:

$$P(t_q|t_d)_{FuzzySet} = \begin{cases} 0 & \text{if } t_d < a_1, \\ f_1(t_d) & \text{if } t_d \geq a_1 \wedge t_d \leq a_2, \\ 1 & \text{if } t_d > a_2 \wedge t_d \leq a_3, \\ f_2(t_d) & \text{if } t_d > a_3 \wedge t_d \leq a_4, \\ 0 & \text{if } t_d > a_4, \end{cases} \tag{5.9}$$

where $f_1(t_d)$ is $\left(\frac{a_1 - t_d}{a_1 - a_2}\right)^n$ if $a_1 \neq a_2$, or 1 if $a_1 = a_2$. $f_2(t_d)$ is $\left(\frac{a_4 - t_d}{a_4 - a_3}\right)^m$ if $a_3 \neq a_4$, or 1 if $a_3 = a_4$.

More recent time-dependent ranking approaches are resorted to a learning-to-rank technique that exploits temporal features (Kanhabua and Nørvåg, 2012; Costa et al., 2014). Kanhabua and Nørvåg (2012) propose a time-sensitive ranking model based on learning-to-rank techniques for explicit temporal queries. To learn the ranking model, they applied two classes of features: temporal and entity-based. For temporal features, both the document focus time and the timestamp are combined. Entity-based features, on the other hand, are used for inferring semantic similarity (such named entities as person, location, or organization). The results show that the SVM^{MAP} learning-to-rank model outperforms a time-aware language modeling approach presented by Berberich et al. (2010).

Similar to previous works on learning multiple ranking models (Bian et al., 2010; Dai et al., 2011), Costa et al. (2014) propose ranking models learned from training data partitioned a document collection by time. They present a temporal-dependent ranking framework that exploits the variance of web characteristics, i.e., document persistence. More precisely, their proposed approach combines the results generated from multiple ranking models using an ensemble method.

A typical ranking problem is to find a function $\mathbf{f}$ with a set of parameters ω that takes query-document feature vectors $\mathcal{X}$ as input and produce a ranking score $\hat{y} = \mathbf{f}(\mathcal{X}, \omega)$. In a learning to rank paradigm, it is aimed at finding the best candidate ranking model $\mathbf{f}^*$ by minimizing a given loss function $\mathbf{L}$ calculated as:

$$\mathbf{f}^* = \arg\min_f \sum_q \mathbf{L}(\hat{y_q}, y_q). \tag{5.10}$$

In order to learn multiple ranking models, a training dataset is constructed from different time periods, thus producing a set of ranking models $\mathbf{M} = \{M_{t_1}, \ldots, M_{t_m}\}$, where t_i is a time period of interest. An ensemble method combines results from different ranking models based on the probability $P(t_i|q)$ that a query candidate q belongs to a time period t_i.

$$\mathbf{f}^* = \arg\min_f \sum_q \mathbf{L}(P(t|d)\hat{y_q}, y_q). \tag{5.11}$$

The ranking objective is to minimize the global relevance loss function, which evaluates the overall training error, instead of assuming the independent loss functions and not considering the correlation between models. Specifically, multiple ranking functions can be computed as:

$$\mathbf{f_1^*}, ..., \mathbf{f_n^*} = \arg\min_{f_1,\ldots,f_n} \sum_q \mathbf{L}(\sum_{j=1}^{n} \hat{y_q}, y_q), \tag{5.12}$$

where n is the number of time-dependent ranking models. After learning different time-dependent models, the final ranking score is produced using an unsupervised ensemble method, i.e., summing the scores generated by all ranking models.

5.3 Event and Entity-aware Ranking

Entity retrieval moves beyond traditional document retrieval to more fine-grained information. In the digital library community, preliminary work (Strötgen and Gertz, 2012) has explored the notion of event-based retrieval, i.e., returning events instead of documents. However, the mentioned work is limited in various aspects: 1) use of a very simple notion of events, 2) evaluate with a small subset of Wikipedia documents, 3) ignore elaborate event detection and indexing techniques, and 4) apply a simple ranking mechanism, i.e., returning results chronologically. In more recent work, Shan et al. (2012) propose a system for event discovery and retrieval in multiple data sources, namely, news articles, videos, and micro-blog streams. The system makes use of a simplified assumption about the time of an event, i.e., employ a clustering approach on a monthly-partitioned sub-collection. An unrealistic assumption is that each document can be linked to just one particular event.

The work by Baeza-Yates (2005) propose to extract temporal expressions from news, index news articles together with temporal expressions, and retrieve temporal information (in this case, future-related events) by using a probabilistic model. A document score is given by multiplying a *keyword* similarity and a time confidence, i.e., a probability that the document's events will actually happen. We can view the confidence as the uncertainty of time. Besides, this work allows a user to explicitly specify temporal information needs, but only on a year-level granularity. Kanhabua et al. (2011) exploit the document features (i.e., *term*, *entity*, *topic* and *time*) for the task of retrieving and ranking the piece of text in the document that contains information about the future events. They concluded that *temporal* features, alongside with *entity-based* features, play an important role in the ranking task.

5.4 Summary

We have presented state-of-the-art time-aware retrieval and ranking methods that incorporate different features (time, and entities) into the retrieval model, and enhance search results, e.g. clustering of based on time and entities. In other words, time-aware retrieval and ranking solutions are achieved by taking into consideration two important aspects. First, they explicitly model the information about time and entity-based annotations, and also consider inherent uncertainties of time into ranking. Second, when a new event has emerged, it needs to dynamically rebuild a ranking model in order to re-rank old documents (or retrieved objects) with respect to the relevance of the new event. We have discussed existing works on time-aware ranking with respect to two main objectives: 1) recency-based ranking, and 2) time-dependent ranking. To this end, we presented recent works that have explored the notion of event-based retrieval, i.e., returning events instead of documents. The limitation of the current approaches is that they make use of a simplified assumption about an event, i.e., employ a clustering approach on a monthly-partitioned sub-collection. A practical assumption is that each document can be linked to just one particular event. One direction to advance event-based retrieval methods is to model and incorporate events into retrieval model and make them the first class citizen of retrieval.

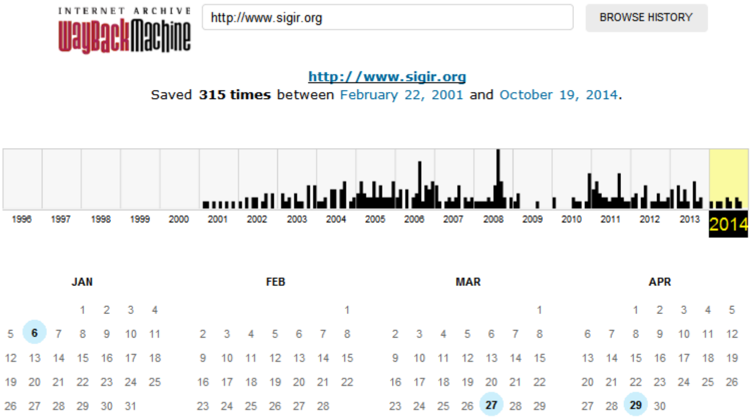
6

Applications of Temporal Information Retrieval

In this section, we provide a short overview of interesting applications on various aspects including existing temporal search engines, temporal analytics and exploration, temporal summarization, temporal clustering of search results, and future event retrieval and prediction.

6.1 Existing Temporal Search Engines

An enormous amount of information is stored in web archives including web pages harvested and added into the archive repository when recrawling, as well as focused web warehouses like news archives. Information in such document repositories are useful for both expert users, e.g., historians, librarians, and journalists, as well as a general user searching for information needs in old versions of web pages. To date, there are existing search systems that provide accessibility to web archives. Unfortunately, current web archive search systems have some shortcomings as illustrated in the following examples.



http://www.sigir.org

http://www.sigir.org
Saved **315 times** between February 22, 2001 and October 19, 2014.

JAN

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

FEB

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

MAR

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

APR

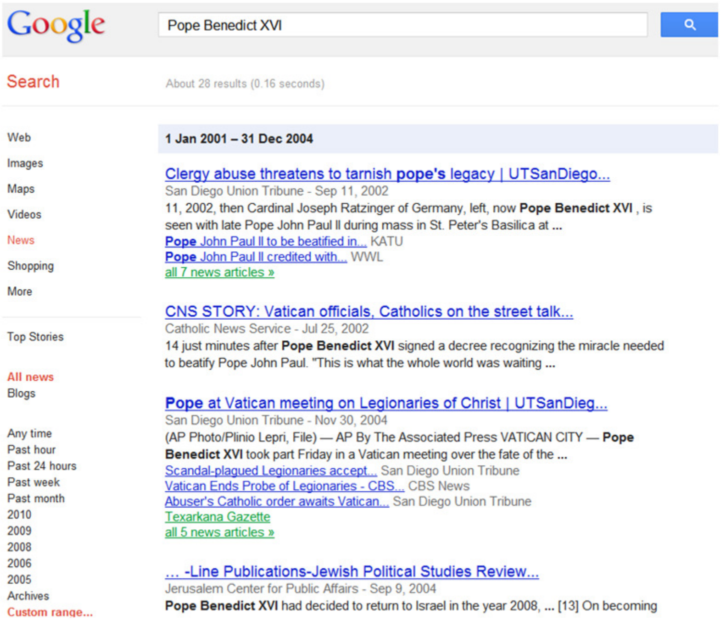
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Figure 6.1: Search results of the query URL http://www.sigir.org in a calendar view (retrieved 2014/12/03).

The Wayback Machine

The Wayback Machine[1] (see Figure 6.1) is a web archive search tool that is provided by the Internet Archive. The Internet Archive is a non-profit organization with the goal of preserving digital document collections as cultural heritage and making them freely accessible online. The Wayback Machine provides the ability to retrieve and access web pages stored in a web archive, and it requires a user to represent her information need by specifying the URL of a web page to be retrieved. For example, given the query URL `http://www.sigir.org`, the results of retrieval are displayed in a calendar view as depicted in Figure 6.1, which displays the number of times the URL `http://www.sigir.org` was crawled by the Wayback Machine (not how many times the site was actually updated). Two major problems of using the Wayback Machine are observable. First, it is inconvenient for a user to specify a URL as a query. Second, there is no easy way to sort search results returned by the tool because the results are displayed in a timeline according to their crawled dates.

[1] `http://archive.org/web/`



About 28 results (0.16 seconds)

1 Jan 2001 – 31 Dec 2004

Clergy abuse threatens to tarnish **pope's** legacy | UTSanDiego...
San Diego Union Tribune - Sep 11, 2002
11, 2002, then Cardinal Joseph Ratzinger of Germany, left, now **Pope Benedict XVI** , is seen with late Pope John Paul II during mass in St. Peter's Basilica at **...**
Pope John Paul II to be beatified in... KATU
Pope John Paul II credited with... WWL
all 7 news articles »

CNS STORY: Vatican officials, Catholics on the street talk...
Catholic News Service - Jul 25, 2002
14 just minutes after **Pope Benedict XVI** signed a decree recognizing the miracle needed to beatify Pope John Paul. "This is what the whole world was waiting **...**

Pope at Vatican meeting on Legionaries of Christ | UTSanDieg...
San Diego Union Tribune - Nov 30, 2004
(AP Photo/Plinio Lepri, File) — AP By The Associated Press VATICAN CITY — **Pope Benedict XVI** took part Friday in a Vatican meeting over the fate of the **...**
Scandal-plagued Legionaries accept... San Diego Union Tribune
Vatican Ends Probe of Legionaries - CBS... CBS News
Abuser's Catholic order awaits Vatican... San Diego Union Tribune
Texarkana Gazette
all 5 news articles »

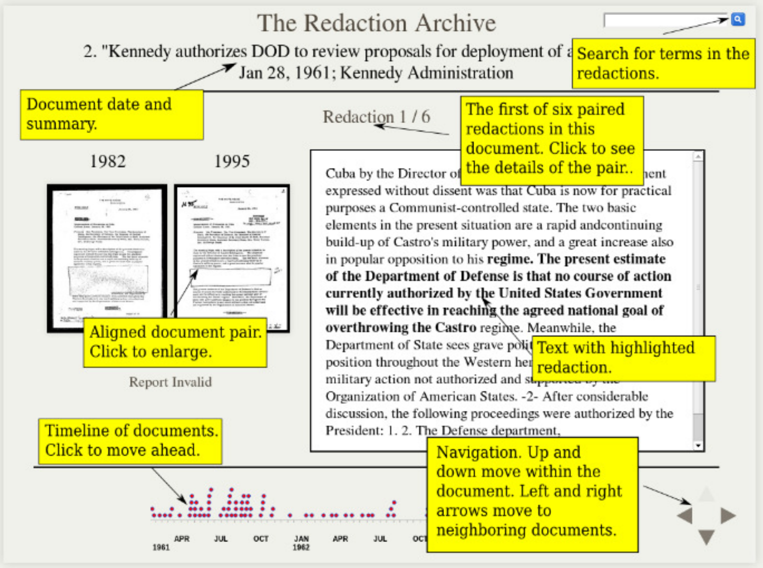
... -Line Publications-Jewish Political Studies Review...
Jerusalem Center for Public Affairs - Sep 9, 2004
Pope Benedict XVI had decided to return to Israel in the year 2008, **...** [13] On becoming

Figure 6.2: Results of the query Pope Benedict XVI and the temporal criteria 2001/01/01 TO 2004/31/12 (retrieved 2012/10/14).

Google News Archive Search

The Google News Archive Search[2] (Figure 6.2) tool allows a user to search a news archive using a keyword query and a date range. In addition, the tool provides the ability to rank search results by relevance or date. However, there is a problem that has not been addressed by this tool yet, e.g., the effect of terminology evolution. Consider the following example; a user wants to search for news about Pope Benedict XVI that are written before 2005. So, the user issues the query Pope Benedict XVI and specifies the temporal criteria 2002/01/01 to 2004/31/12. As shown in Figure 6.2, only a small number of documents are returned by the tool where most of them are not relevant to the Pope Benedict XVI. In other words, this problem can be viewed as vocabulary mismatch caused by the fact that the term Pope Benedict XVI was not widely used before 2005/04/19 (the date when his papacy began).

[2] `http://news.google.com/archivesearch`



The Redaction Archive

2. "Kennedy authorizes DOD to review proposals for deployment of a

Jan 28, 1961; Kennedy Administration

Redaction 1 / 6

1982 1995

Report Invalid

Cuba by the Director of ... ment expressed without dissent was that Cuba is now for practical purposes a Communist-controlled state. The two basic elements in the present situation are a rapid andcontinuing build-up of Castro's military power, and a great increase also in popular opposition to his **regime. The present estimate of the Department of Defense is that no course of action currently authorized by the United States Government will be effective in reaching the agreed national goal of overthrowing the Castro** regime. Meanwhile, the Department of State sees grave polit... position throughout the Western he... military action not authorized and s... Organization of American States. -2- After considerable discussion, the following proceedings were authorized by the President: 1. 2. The Defense department,

Search for terms in the redactions.

Document date and summary.

The first of six paired redactions in this document. Click to see the details of the pair..

Aligned document pair. Click to enlarge.

Text with highlighted redaction.

Timeline of documents. Click to move ahead.

Navigation. Up and down move within the document. Left and right arrows move to neighboring documents.

Figure 6.3: Documents about Kennedy are redacted on different dates and aligned on timeline.

The Redaction Archive

The declassification engine[3] (see Figure 6.3) refers to a set of applications that will provide new forms of access to official documents as well as tools to help interpret them. Currently its main projects are the (de)classifier, a tool displaying cable activity over time for different embassies and topics contrasting the number of documents declassified with the number still withheld, the (de)sanitizer, an app that illuminates previously redacted text to show what kinds of information is considered particularly sensitive, and the sphere of influence, a visualization of diplomatic activity around the globe based on the volume of classified and declassified cable traffic. In addition, searching in this unique longitudinal collection is somehow different from searching over web pages especially due to the huge level of redundancy that rises from near-identical content or web pages that have been crawled all over again.

[3] http://www.wired.com/2013/05/the-declassification-engine/

6.2 Analysis and Exploration over Time

Some applications have taken time-based exploration of textual archives beyond just searching over time. Filtering and displaying information might benefit from presenting time information conveniently in some domains. For instance, Koen and Bender (2000) enrich the news reading experience by augmenting articles using time information automatically extracted from documents.

Matthews et al. (2010) describe Time Explorer, a system that provides time-line based exploration of news archives. Time Explorer combines a number of interesting features present in other time-based systems, although extended in several important ways. First, users are enticed to discover how entities such as people and locations associated with a query change over time. Second, by searching on time expressions extracted automatically from text, the application allows the user to explore not only how topics evolved in the past, but also how they will continue to evolve in the future. All these features are combined in an intuitive easy-to-use interface, which is always a great challenge when designing search engines that allow for extended capabilities.

Other exploration-based systems have turned into other sources of textual information, for instance word evolution over time. This is naturally promising research direction, since there are available digitalized collections that span centuries. Odijk et al. (2012) present a system that combines faceted search techniques and interactive visualization in order to help to gain a deeper understanding of the relevant parts of the collection. In their system, the time axis plays a central role all over the user interface, allowing to modify the different results visualizations by selecting different time slices.

These systems need to employ a combination of several time-aware algorithms. For instance, they will require to extract time-related information from a given underlying textual collection, index this information along with the standard collection's contents and perform a combination of temporal query analysis and ranking after a user query is posed.

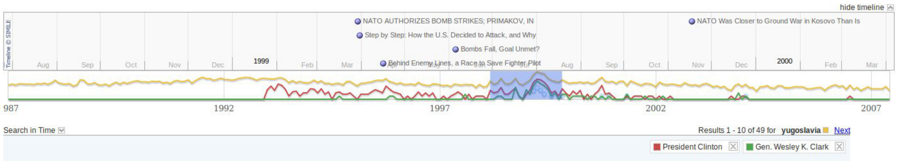

hide timeline
Timeline © SIMILE
NATO AUTHORIZES BOMB STRIKES; PRIMAKOV, IN
Step by Step: How the U.S. Decided to Attack, and Why
Bombs Fall, Goal Unmet?
Behind Enemy Lines, a Race to Save Fighter Pilot
NATO Was Closer to Ground War in Kosovo Than Is
Aug
Sep
Oct
Nov
Dec
1999
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2000
Feb
Mar
987
1992
1997
2002
2007
Search in Time
Results 1 - 10 of 49 for yugoslavia
Next
President Clinton
Gen. Wesley K. Clark

Figure 6.4: Capture of TimeExplorer's timeline navigation interfaced, for the query "yugoslavia" and two related entities (in red and green), along with supporting result documents.

6.3 Temporal Summarization

Another stream of applications has focused into exploiting the use of time for enhanced story telling. The task of news summarization has been studied previously ranging from multi-document summarization (Erkan and Radev, 2004) to generating a timeline summary for a specific news story (Yan et al., 2011; Tran et al., 2013; Zhao et al., 2013; McCreadie et al., 2014). One of those applications intertwines entity retrieval, which is now commonly present in a wide range of commercial applications, (Demartini et al., 2010) with a timeline-style summary. The working example would look as follows. A user enters a topic into a news search engine and obtains a list of relevant results, ordered by time. Furthermore, the user subscribes to this query so in the future she will continue to receive the latest news on this query. The time dimension comes into play when the user is observing a current document, and one may want to show the most relevant entities of the document for her query taking into account features extracted from previous documents.

Sipos et al. (2012) present another form of temporal summary in terms of landmark documents, authors, and topics. They explicitly model how documents temporary influence each other and cast the summarization problem as a coverage problem over words anchored in time. The resulting system is able to provide informative and non-redundant summaries over time.

Arguably, the most widespread summarization technology is the (query) focused summaries produced by search engines, or search results snippets. Those are useful to assist users in deciding whether a document is relevant for a query or not. In this line, Svore et al. (2012)

explored the value of biasing search result snippets towards new webpage content. This is an important point, given that a large portion of the web content is highly dynamic. Their results indicate that users find useful temporal information in search snippets for trending queries but not for general queries. Additionally, this is more valuable for pages that have not been recently crawled. Alonso et al. (2009a, 2011a) present a method to augment search snippets using temporal expressions, which include the most frequent units of time appearing in search result pages. A crowsourced evaluation showed that around 90% of the users found those snippets to be useful when combined with standard document surrogates. Future research in this area could dig deeper into what classes of queries are more amenable to temporal snippets and if those can be identified automatically.

6.4 Temporal Clustering of Search Results

Another popular application that makes use of time-based IR techniques is search results clustering, which is an important feature for some information retrieval applications, in particular, enterprise search systems. Alonso and Gertz (2006) describe a prototype that is able to display date and time attributes per cluster. Those attributes are extracted from the textual content of documents that belong to a particular cluster.

Furthermore, Alonso et al. (2009b) extend the idea of reusing temporal information embedded in documents to enhance results presentation by introducing a timeline-based display of results. The timelines span different time granularities and display temporal information extracted automatically from documents and made explicit. They also explore how search results can be clustered according to time and how to produce temporal snippets to navigate through documents returned.

Campos et al. (2012b) describe a method to group search result documents at a year level using a similarity measure that identifies the most relevant dates. Each group is then displayed differently in the results page, allowing for an easier exploration of the search results, as demonstrated by a user survey. In general, new interfaces that exploit mining of temporal information is an active area of research.

6.5 Future Event Retrieval and Prediction

An exciting array of temporal applications are the ones based on *prediction of events*, ranging from entertainment (when a movie is going to be popular) to natural disasters or even football games outcomes (UzZaman et al., 2012).

Kanhabua et al. (2011) report that nearly one third of news articles contain references to future events. This information is not easily accessible, although it can be crucial to understanding news stories and how events will develop for a given topic. Kanhabua et al. (2011) propose a new task to address the problem of retrieving and ranking sentences that contain mentions to future events, which they call ranking related news predictions, and engineer solutions based on term, topic and temporal similarities. In a similar vein, Jatowt and Au Yeung (2011) describe a method to extract and summarize future-related information which, in principle, could allow people to produce a collective image of how the future will look like, and be prepared for future events. The model itself employs clustering for detecting future events, time and topic-based similarities. Conversely, Au Yeung and Jatowt (2011) show that a careful analysis of references to the past in news articles provide deep insight into the collective memories and societal views of different countries.

In their recent work, Kanhabua et al. (2014) have presented a study of collective memories in Wikipedia. In more detail, they study the trigger of revisiting behavior for a large set of event pages by exploiting page views and time series analysis, as well as identifying of most important catalyst features. This provides a systematic analysis of the memory of past events (what is remembered), which complement the news coverage perspective of the work by Au Yeung and Jatowt (2011). This sort of studies can be applied to other domains apart from news articles, like social media, which contain messages that often reflect expectations or memories of social media users. Jatowt et al. (2015) found out that social media is mostly using for referring to *present* events and the patterns of temporal attention are stable apart from a few exceptions (for instance, New Year's eve).

The most challenging task is the actual *forecast* of forthcoming events. Amodeo et al. (2011b) present a method to turn a standard

IR engine into a future event predicting machine. The system operates on a per query basis and it makes use of the publication dates of the retrieved documents to capture trends and periodicity of associated events. Future burst for the query are then estimated using the periodicity of historic data by intertwining a probabilistic and time-series model.

Radinsky and Horvitz (2013) describe how to forecast real-life events by extracting information from a corpus comprised of 22 years of news stories. In short, the method generalizes sequences of events from multiple information sources and it is able to identify the increases in likelihood of disease outbreaks, deaths and riots.

7

Conclusions and Outlook

With the growing volume of and reliance on digital documents, there is a clear need for approaches to keep relevant information accessible, available, and meaningful over time. In the past decade, time-aware information retrieval has emerged as a topic revolving around the fundamental question of how the time dimension affects search processes. In this survey, we have introduced the general topic of web evolution and pinpointed a number of issues related to the temporal aspects of IR. In particular, we have addressed the impacts of this evolution on gathering and dynamically processing collections of documents that span and change over different periods of time, temporal query analysis, as well as time-aware retrieval and ranking. We wrapped up with a review of some recent applications in related research areas that the time dimension is considered to be a critical role. To this end, we also provide an overview of standard and test collections available for conducting research and evaluation of the concepts, methods and algorithms developed in this research area in the appendix.

There are still several open issues and opportunities for further research. In the following, we outline some of the promising topics for future research.

Real-time web mining. Social network services, e.g., Twitter, blogs, and discussion forums, have gained increasing interests. The contents generated from social networks have increased dramatically, and challenges when dealing with such data include real-time (stream) data, noisy texts (unedited language), and dynamic topics/events. An interesting research direction could involve time-aware approaches to *mine user-generated content.* One example of possible applications is Online Reputation Management that is created to monitor social media in order to detect contents or opinions about products, people and organizations (Spina et al., 2014). Recently, there have been several initiatives in leveraging the social Web for large scale event management. Numerous real-time web applications increasingly use Twitter for real-time analysis tasks, for instance, detecting natural disasters (Sakaki et al., 2010), political persuasion (Lumezanu et al., 2012; Sang and Bos, 2012), or current trends (Diaz-Aviles et al., 2012; Lampos and Cristianini, 2012).

Spatio-temporal search. Within the scope of this survey, we focus on queries containing temporal information needs. In some cases queries may contain not only time, but also geographic information. More precisely, a user may search for a particular event or a local business in which location and time will impact results relevance. This line of research has gained increasing interest. For example, Mishra et al. (2014) study the problem of *time-critical search* that incorporate geospatial information for accurately detecting urgent information needs, given a query and a diverse set of other features, such as, spanning topical, temporal, behavioral. In the context of social media, Li and Sun (2014) focus on extracting fine-grained locations mentioned in tweets with temporal awareness. More precisely, they extract each point-of-interest (POI) mention in a tweet and predict whether the user has visited, is currently at, or will soon visit this POI.

Brain-inspired information access. Another interesting direction to advance research in this area is brain-inspired approaches, which embrace human remembering and forgetting into long-term information access. Niederée et al. (2015) envision a concept of *managed forgetting* for systematically dealing with information that progressively ceases in importance and with redundancy – leading to a form of *Forgetful Digital*

Memory. Managed forgetting is meant to complement rather than copy human remembering and forgetting. It can be regarded as functions of attention and significance dynamics, which ensure that important content is kept safe, useful and understandable over time. In the conceptual model proposed by Niederée et al. (2015), the value of information is multi-faceted and can be considered from different perspectives, namely, (1) the short-term importance of information (e.g., short-term interests by analyzing the observed activity logs (White et al., 2010)), and (2) the long-term importance of a resource (e.g., expected usefulness by estimating a probability of future use for the resource (Ceroni et al., 2015)).

Finally, it should be noted that aspects of temporal information retrieval will also be an important component in related research areas like recommender systems (Stefanidis et al., 2013), temporal knowledge bases (Hoffart et al., 2013), and expertise search (Rybak et al., 2014). We expect temporal information retrieval to continue to be a research area with high activity in the coming years, and this is also the main motivation for writing this survey/tutorial: to make it easier for other researchers to get acquainted to this important research area.

Appendices

A

Research Resources

For evaluation of temporal information retrieval techniques, research resources such as temporal document collections are needed. Standard document collections made for use by, e.g., the Text REtrieval Conference (TREC) or Conference and Labs of the Evaluation Forum (CLEF) are mostly not suitable for this task. For example, the TREC ad-hoc test collection covers documents from 1987 to 1994, where the topics of documents are somewhat old when we are interested in both old and contemporary stories. A more recent collection, the ad-hoc track of CLEF 2008, is composed of recent documents from 2001 to 2006, but only covers 5-6 years period. Thus, these collections are often not applicable for evaluating the long-term aspects of search.

In the following, we provide a description of a number of temporal document collections that are available for conducting research in temporal information retrieval: The Times Archive, New York Times Annotated Corpus, Wikipedia, and collections based on user-generated content. We also present resources that have been made available as part of research challenges like, e.g., TREC, and as examples we present two recent challenges related to temporal information retrieval, namely Temporalia and the TREC Temporal Summarization Track.

The Times Archive

The Times Archive[1] consists of approximately 20 million news articles available in full text spanning from year 1785 to 1985, which results in over 200 years of data. Starting with almost 4400 articles from 1785, the number of articles increases steadily during the first 100 years. In the early 20th century the increase becomes more rapid and in 1911 we have almost double the number of articles as in 1905. All articles in The Times Archive have been manually classified into 18 categories (e.g., news, sport, obituaries, and politics and parliament) during the digitization process.

New York Times Annotated Corpus

The New York Times Annotated Corpus[2] contains over 1.8 million articles covering a period from January 1987 to June 2007. There are more than 650,000 article summaries written by library scientists, and over 1.5 million articles were manually tagged from a vocabulary of people, organizations and locations using a controlled vocabulary that is applied consistently across the collection. For instance, if one article mentions “Bill Clinton” and another refers to “President William Jefferson Clinton”, both articles will be tagged with “CLINTON, BILL”. In addition, there are over 275,000 algorithmically-tagged articles that have been hand verified by the online production staff at nytimes.com. This annotation data can served as ground truths for problems such as the effect of named entity evolution in searching web archives. Thus, the NYT Annotated Corpus is a valuable resource for conducting research on a number of topics related to temporal information retrieval. There are several works that use this temporal collection as a test collection in a wide range of applications, e.g., (Kanhabua et al., 2011; Berberich et al., 2010; Radinsky and Horvitz, 2013).

[1] `http://archive.timesonline.co.uk/tol/archive/`

[2] `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2008T19`

Temporal Wikipedia

The English version of Wikipedia provides more than 4 millions articles with the complete history of more than 500 millions edits since its launch in 2001. This represents an important collection of human-intended documents with an average of 19 edits per page.

The complete history of Wikipedia can be used to enable temporal retrieval and analysis over Wikipedia articles, at all points in time since 2001. Unfolding these edits to provide access to arbitrary temporal snapshots and sequences of snapshots of Wikipedia since 2001. This setup provides us with a web-archive-like collection as a good testbed for entity extraction and evolution, as well as for time- and entity-aware retrieval, with a relatively simple, consistent and text-based collection.

In addition to the raw Wikipedia data, there are also semantic knowledge bases like DBpedia [3] that are created based on information in Wikipedia. The content of DBpedia is extracted from Wikipedia infoboxes and tables, and describes over 4.5 million entities. More than 4.2 million of these entities have been matched in a collective effort to a well-defined ontology, including persons, places, creative works, organizations, species, and diseases.

Previous works on analyzing Wikipedia history are, for example, (Kanhabua and Nørvåg, 2010b; Georgescu et al., 2013).

User-generated Content Collections

Temporal search and analytics in user-generated content, e.g., blogs and social media, have been gaining increasing research interest. The TREC Blog data[4] covers 1.3 million blogs which were followed over a time period of over 1 year, which amount to approximately 2.3 TB of data. Nguyen and Kanhabua (2014) leverage this data collection as an external source for mining dynamic subtopics for web queries.

Flickr, a social photo sharing platform, comprises metadata for approximately 170 million photos, which includes titles, tags, time stamps, geo-tags, descriptions, etc. An alternative source of data gathering is the

[3] http://wiki.dbpedia.org/Datasets

[4] http://ir.dcs.gla.ac.uk/wiki/TREC-BLOG

Spinn3r service which collects and offers Web 2.0 content such as, blogs, news and social media, for analytic companies, search engines, and research. This service monitors around 40 million sources and generates approximately 1,200,000 posts per hour (around 200GB per day).

Evaluation Workshops

There have been a number of research challenges related to temporal IR. These tasks changes frequently; in the following we will present two such challenges that have been organized recently: the TREC Temporal Summarization Track and Temporalia (Temporal Information Access).

The TREC Temporal Summarization track[5] relates to event detection and filtering. It is composed of two subtasks: Sequential Update Summarization and Value Tracking. The task of Sequential Update Summarization is to find timely, sentence-level, reliable, relevant and non-redundant updates about a developing event. The value tracking task aims at tracking values of event-related attributes that are of high importance to the event like the number of fatalities during a natural disaster or financial impact of a corporate decision.

The Temporalia challenge (Joho et al., 2014a,b, 2013) possibly comprises the first test collection geared to measure the performance of search applications across temporal information needs categories in a systematic way. Temporalia comes with a set of documents and topics along with relevance judgments for two different tasks and establishes common grounds for designing and analyzing temporally-aware information access systems. The document collection employed in the Temporalia challenge comprises a selected number of sites which have been crawled on a daily basis from May 2011 to March 2013, which makes it suitable for evaluating tasks that require web snapshots across different periods of time.

Temporalia, like other evaluation campaigns, has used a crowdsourced platform for gathering relevance judgments in a relatively inexpensive manner. Provided that appropriate quality control mechanisms are put in place, e.g., spam detection and label aggregation, platforms

[5] http://www.trec-ts.org/

like Amazon's Mechanical Turk and Crowdflower bring effective ways to overcome the lack of suitable data sources for new research tasks.

The size of the document collection is approximately 20GB and contains around 3.8 million documents collected from about 1,500 different blogs and news sources. Further, this collection also provides three kinds of annotations: sentence splitting, named entities, and time annotations.

There are two tasks addressed within Temporalia: temporal query intent classification (TQIC) and temporal information retrieval (TIR).

TQIC consists of classifying queries into a predetermined set of temporal classes based on their implicit or explicit temporal intent. Queries can be categorized into the following classes: *past* (queries whose search results are not expected to change much with time), *recency* (whose search results are expected to be timely and up-to-date), and *future* (about predicted or scheduled events) and *atemporal* (without any clear temporal intent). The data includes 400 queries manually annotated with their respective temporal classes. Approaches to TQIC consisted, generally, in machine learning classifiers that use term and POS-ngrams along with more sophisticated linguistic features like named entities or verb tenses.

TIR has the goal to retrieve a set of documents in response to a search topic that incorporates a time factor. In addition to a standard TREC-like search topic description, a TIR topic contains query submitting date information to indicate the relationship between the query and searcher. The data contains 36K relevance judgments graded as not relevant, relevant and highly relevant over 50 queries. Queries are split into four types, i.e., each one of those 50 queries contain 4 subtopics one per temporal class (past, recency, future, atemporal) and the task consists to not only retrieve documents relevant to the query but they are also required to belong to that particular temporal class. The approaches to TIR were varied in nature. One popular technique was to learn to re-ranking a BM25 retrieved set of document for a given query using learning to rank and temporal expressions extracted from the documents' contents as features, along with linguistic features like verb tenses. Other approaches tried to estimate temporal and topical relevance, and then combine them for a final document and class score.

References

Adar, E., J. Teevan, S. T. Dumais, and J. L. Elsas (2009), 'The Web Changes Everything: Understanding the Dynamics of Web Content'. In: *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. pp. 282–291.

Aji, A., Y. Wang, E. Agichtein, and E. Gabrilovich (2010), 'Using the Past to Score the Present: Extending Term Weighting Models Through Revision History Analysis'. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 629–638.

Alici, S., I. S. Altingovde, R. Ozcan, B. B. Cambazoglu, and O. Ulusoy (2011), 'Timestamp-based Result Cache Invalidation for Web Search Engines'. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 973–982.

Alici, S., I. S. Altingovde, R. Ozcan, B. B. Cambazoglu, and O. Ulusoy (2012), 'Adaptive Time-to-live Strategies for Query Result Caching in Web Search Engines'. In: *Proceedings of the 34th European Conference on Advances in Information Retrieval*. pp. 401–412.

Allan, J., R. Papka, and V. 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*. pp. 37–45.

Alonso, O., R. Baeza-Yates, and M. Gertz (2009a), 'Effectiveness of Temporal Snippets'. In: *World Wide Web Conference Series*.

Alonso, O. and M. Gertz (2006), 'Clustering of Search Results Using Temporal Attributes'. In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 597–598.

Alonso, O., M. Gertz, and R. Baeza-Yates (2011a), 'Enhancing Document Snippets Using Temporal Information'. In: *Proceedings of the 18th International Conference on String Processing and Information Retrieval*. pp. 26–31.

Alonso, O., M. Gertz, and R. A. Baeza-Yates (2009b), 'Clustering and Exploring Search Results Using Timeline Constructions'. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. pp. 97–106.

Alonso, O., J. Strötgen, R. A. Baeza-Yates, and M. Gertz (2011b), 'Temporal Information Retrieval: Challenges and Opportunities'. In: *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW 2011)*.

Amodeo, G., G. Amati, and G. Gambosi (2011a), 'On Relevance, Time and Query Expansion'. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pp. 1973–1976.

Amodeo, G., R. Blanco, and U. Brefeld (2011b), 'Hybrid Models for Future Event Prediction'. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pp. 1981–1984.

Anand, A., S. Bedathur, K. Berberich, and R. Schenkel (2010), 'Efficient Temporal Keyword Search over Versioned Text'. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 699–708.

Anand, A., S. Bedathur, K. Berberich, and R. Schenkel (2011), 'Temporal Index Sharding for Space-time Efficiency in Archive Search'. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 545–554.

Anick, P. G. and R. A. Flynn (1992), 'Versioning a Full-text Information Retrieval System'. In: *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 98–111.

Arun, R., V. Suresh, C. E. Veni Madhavan, and M. N. Narasimha Murthy (2010), 'On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations'. In: *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*. pp. 391–402.

Au Yeung, C.-m. and A. Jatowt (2011), 'Studying How the Past is Remembered: Towards Computational History Through Large Scale Text Mining'. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pp. 1231–1240.

Baeza-Yates, R. A. (2005), 'Searching the Future'. In: *Proceedings of SIGIR workshop on mathematical/formal methods in information retrieval MF/IR*.

Baeza-Yates, R. A. and B. A. Ribeiro-Neto (2011), *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England.

Bar-Yossef, Z. and N. Kraus (2011), 'Context-sensitive Query Autocompletion'. In: *Proceedings of the 20th International Conference on World Wide Web*. pp. 107–116.

Barbosa, L., A. C. Salgado, F. de Carvalho, J. Robin, and J. Freire (2005), 'Looking at Both the Present and the Past to Efficiently Update Replicas of Web Content'. In: *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*. pp. 75–80.

Beitzel, S. M., E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman (2007), 'Temporal Analysis of a Very Large Topically Categorized Web Query Log'. *J. Am. Soc. Inf. Sci. Technol.* **58**(2), 166–178.

Beitzel, S. M., E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder (2004), 'Hourly Analysis of a Very Large Topically Categorized Web Query Log'. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 321–328.

Berberich, K., S. Bedathur, T. Neumann, and G. Weikum (2007), 'A Time Machine for Text Search'. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 519–526.

Berberich, K., S. J. Bedathur, O. Alonso, and G. Weikum (2010), 'A Language Modeling Approach for Temporal Information Needs'. In: *Proceedings of the 32nd European Conference on IR Research on Advances in Information Retrieval*. pp. 13–25.

Berberich, K., S. J. Bedathur, M. Sozio, and G. Weikum (2009), 'Bridging the Terminology Gap in Web Archive Search'. In: *Proceedings of the 12th International Workshop on the Web and Databases (WebDB)*.

Berberich, K., M. Vazirgiannis, and G. Weikum (2005), 'Time-Aware Authority Ranking'. *Internet Mathematics* **2**(3).

Berry, M. (2003), *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer.

Beyer, H. and K. Holtzblatt (1998), *Contextual Design: Defining Customer-centered Systems*. Morgan Kaufmann Publishers Inc.

Bian, J., X. Li, F. Li, Z. Zheng, and H. Zha (2010), 'Ranking Specialization for Web Search: A Divide-and-conquer Approach by Using Topical RankSVM'. In: *Proceedings of the 19th International Conference on World Wide Web*. pp. 131–140.

Blanco, R., E. Bortnikov, F. Junqueira, R. Lempel, L. Telloli, and H. Zaragoza (2010), 'Caching Search Engine Results over Incremental Indices'. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 82–89.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003), 'Latent dirichlet allocation'. *J. Mach. Learn. Res.* **3**, 993–1022.

Bøhn, C. and K. Nørvåg (2010), 'Extracting Named Entities and Synonyms from Wikipedia'. In: *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications AINA'2010*. pp. 1300–1307.

Boldi, P., B. Codenotti, M. Santini, and S. Vigna (2004), 'UbiCrawler: A Scalable Fully Distributed Web Crawler'. *Softw. Pract. Exper.* **34**(8), 711–726.

Box, G. E. P. and G. Jenkins (1990), *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated.

Broder, A. Z., N. Eiron, M. Fontoura, M. Herscovici, R. Lempel, J. McPherson, R. Qi, and E. J. Shekita (2006), 'Indexing Shared Content in Information Retrieval Systems'. In: *Proceedings of the 10th International Conference on Extending Database Technology*. pp. 313–330.

Cambazoglu, B. B., F. P. Junqueira, V. Plachouras, S. Banachowski, B. Cui, S. Lim, and B. Bridge (2010), 'A Refreshing Perspective of Search Engine Caching'. In: *Proceedings of the 19th International Conference on World Wide Web*. pp. 181–190.

Campos, R., G. Dias, A. Jorge, and C. Nunes (2012a), 'GTE: a Distributional Second-Order Co-Occurrence Approach to Amprove the Identification of Top Relevant Dates in Web Snippets'. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 2035–2039.

Campos, R., G. Dias, A. Jorge, and C. Nunes (2014a), 'GTECluster: A Temporal Search Interface for Implicit Temporal Queries'. In: *Proceedings of the 36th European Conference on IR Research*. pp. 775–779.

Campos, R., G. Dias, A. M. Jorge, and A. Jatowt (2014b), 'Survey of Temporal Information Retrieval and Related Applications'. *ACM Comput. Surv.* **47**(2), 15:1–15:41.

Campos, R., G. Dias, A. M. Jorge, and C. Nunes (2014c), 'GTE-Rank: Searching for Implicit Temporal Query Results'. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. pp. 2081–2083.

Campos, R., A. M. Jorge, G. Dias, and C. Nunes (2012b), 'Disambiguating Implicit Temporal Queries by Clustering Top Relevant Dates in Web Snippets'. In: *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*. pp. 1–8.

Cao, J., T. Xia, J. Li, Y. Zhang, and S. Tang (2009), 'A Density-Based Method for Adaptive LDA Model Selection'. *Neurocomput.* **72**(7-9), 1775–1781.

Carmel, D. and E. Yom-Tov (2010), *Estimating the Query Difficulty for Information Retrieval*. Morgan & Claypool Publishers.

Carterette, B. and P. Chandar (2009), 'Probabilistic Models of Ranking Novel Documents for Faceted Topic Retrieval'. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. pp. 1287–1296.

Ceroni, A., V. Solachidis, C. Niederée, O. Papadopoulou, N. Kanhabua, and V. Mezaris (2015), 'To Keep or not to Keep: An Expectation–oriented Photo Selection Method for Personal Photo Collections'. In: *Proceedings of International Conference on Multimedia Retrieval*.

Chambers, N. (2012), 'Labeling Documents with Timestamps: Learning from Their Time Expressions'. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. pp. 98–106.

Chang, A. X. and C. Manning (2012), 'SUTime: A library for recognizing and normalizing time expressions'. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation*. pp. 3735–3740.

Chang, F., J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber (2006), 'Bigtable: A Distributed Storage System for Structured Data'. In: *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*. pp. 15–15.

Cheng, S., A. Arvanitis, and V. Hristidis (2013), 'How Fresh Do You Want Your Search Results?'. In: *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*. pp. 1271–1280.

Cho, J. and H. Garcia-Molina (2000), ‘The Evolution of the Web and Implications for an Incremental Crawler’. In: *Proceedings of the 26th International Conference on Very Large Data Bases*. pp. 200–209.

Cho, J. and H. Garcia-Molina (2003a), ‘Effective Page Refresh Policies for Web Crawlers’. *ACM Trans. Database Syst.* **28**(4), 390–426.

Cho, J. and H. Garcia-Molina (2003b), ‘Estimating Frequency of Change’. *ACM Trans. Internet Technol.* **3**(3), 256–290.

Ciglan, M. and K. Nørvåg (2010), ‘WikiPop: Personalized Event Detection System Based on Wikipedia Page View Statistics’. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 1931–1932.

Cleveland, R. B., W. S. Cleveland, J. E. McRae, and I. Terpenning (1990), ‘STL: A Seasonal-Trend Decomposition Procedure Based on Loess’. *Journal of Official Statistics* **6**, 3–73.

Cormack, G. V. and M. R. Grossman (2014), ‘Evaluation of Machine-learning Protocols for Technology-assisted Review in Electronic Discovery’. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. pp. 153–162.

Corso, G. M. D., A. Gullí, and F. Romani (2005), ‘Ranking a stream of news’. In: *Proceedings of the 14th international conference on World Wide Web*. pp. 97–106.

Costa, M., F. Couto, and M. Silva (2014), ‘Learning Temporal-dependent Ranking Models’. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. pp. 757–766.

Costa, M., D. Gomes, F. Couto, and M. Silva (2013), ‘A Survey of Web Archive Search Architectures’. In: *Proceedings of the 22nd International Conference on World Wide Web (Companion)*. pp. 1045–1050.

Croft, W. B., D. Metzler, and T. Strohman (2009), *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 1st edition.

Cronen-Townsend, S., Y. Zhou, and W. B. Croft (2002), ‘Predicting Query Performance’. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 299–306.

Dai, N. and B. D. Davison (2010a), ‘Freshness Matters: in Flowers, Food, and Web Authority’. In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. pp. 114–121.

Dai, N. and B. D. Davison (2010b), 'Mining Anchor Text Trends for Retrieval'. In: *Proceedings of the 32nd European Conference on Advances in Information Retrieval*. pp. 127–139.

Dai, N., M. Shokouhi, and B. D. Davison (2011), 'Learning to Rank for Freshness and Relevance'. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 95–104.

Dakka, W., L. Gravano, and P. G. Ipeirotis (2012), 'Answering General Time-Sensitive Queries'. *IEEE Transactions on Knowledge and Data Engineering* **24**(2), 220–235.

de Boer, V., M. van Someren, and B. J. Wielinga (2010), 'Extracting Historical Time Periods from the Web'. *Journal of the American Society for Information Science and Technology* **61**, 1888–1908.

de Jong, F., H. Rode, and D. Hiemstra (2005), 'Temporal Language Models for the Disclosure of Historical Text'. In: *Proceedings of the 16th International Conference of the Association for History and Computing*. pp. 161–168.

Demartini, G., M. M. S. Missen, R. Blanco, and H. Zaragoza (2010), 'Entity Summarization of News Articles'. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 795–796.

Deng, H., I. King, and M. R. Lyu (2009), 'Entropy-biased Models for Query Representation on the Click Graph'. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 339–346.

Diaz-Aviles, E., L. Drumond, L. Schmidt-Thieme, and W. Nejdl (2012), 'Real-time Top-N Recommendation in Social Streams'. In: *Proceedings of the sixth ACM conference on Recommender systems*. pp. 59–66.

Dong, A., Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz (2010a), 'Towards Recency Ranking in Web Search'. In: *Proceedings of the third ACM international conference on Web search and data mining*. pp. 11–20.

Dong, A., R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha (2010b), 'Time is the Essence: improving Recency Ranking using Twitter Data'. In: *Proceedings of the 19th international conference on World wide web*. pp. 331–340.

Dou, Z., R. Song, J.-Y. Nie, and J.-R. Wen (2009), 'Using Anchor Texts with Their Hyperlink Structure for Web Search'. In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 227–234.

Efron, M. (2013), 'Query Representation for Cross-temporal Information Retrieval'. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 383–392.

Efron, M., J. Lin, J. He, and A. de Vries (2014), 'Temporal Feedback for Tweet Search with Non-parametric Density Estimation'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 33–42.

Elsas, J. L. and S. T. Dumais (2010), 'Leveraging Temporal Dynamics of Document Content in Relevance Ranking'. In: *Proceedings of the third ACM international conference on Web search and data mining*. pp. 1–10.

Erkan, G. and D. R. Radev (2004), 'LexRank: Graph-based Lexical Centrality As Salience in Text Summarization'. *J. Artif. Int. Res.* **22**(1), 457–479.

Ernst-Gerlach, A. and N. Fuhr (2006), 'Generating Search Term Variants for Text Collections with Historic Spellings'. In: *Proceedings of the 28th European Conference on IR Research on Advances in Information Retrieval*. pp. 49–60.

Ernst-Gerlach, A. and N. Fuhr (2007), 'Retrieval in text collections with historic spelling using linguistic and spelling variants'. In: *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. pp. 333–341.

Fagni, T., R. Perego, F. Silvestri, and S. Orlando (2006), 'Boosting the Performance of Web Search Engines: Caching and Prefetching Query Results by Exploiting Historical Usage Data'. *ACM Trans. Inf. Syst.* **24**(1), 51–78.

Ferron, M. and P. Massa (2012), 'Psychological Processes Underlying Wikipedia Representations of Natural and Manmade Disasters'. In: *Proceedings of WikiSym '12*.

Fetterly, D., M. Manasse, M. Najork, and J. Wiener (2003), 'A Large-scale Study of the Evolution of Web Pages'. In: *Proceedings of the 12th International Conference on World Wide Web*. pp. 669–678.

Fontoura, M., R. Lempel, R. Qi, and J. Y. Zien (2007), 'Inverted Index Support for Numeric Search'. *Internet Mathematics* **3**(2), 153–185.

Garcia-Fernandez, A., A.-L. Ligozat, M. Dinarelli, and D. Bernhard (2011), 'When was it Written? Automatically Determining Publication Dates'. In: *Proceedings of the 18th international conference on String processing and information retrieval*. pp. 221–236.

Ge, T., B. Chang, S. Li, and Z. Sui (2013), ‘Event-Based Time Label Propagation for Automatic Dating of News Articles’. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pp. 1–11.

Georgescu, M., N. Kanhabua, D. Krause, W. Nejdl, and S. Siersdorfer (2013), ‘Extracting Event-Related Information from Article Updates in Wikipedia’. In: *Proceedings of the 35th European conference on Advances in Information Retrieval*. pp. 254–266.

Goldberg, Y. and J. Orwant (2013), ‘A dataset of syntactic-ngrams over time from a very large corpus of english books’. In: *Proceedings of Second Joint Conference on Lexical and Computational Semantics (* SEM)*, Vol. 1. pp. 241–247.

Gomes, D., J. a. Miranda, and M. Costa (2011), ‘A Survey on Web Archiving Initiatives’. In: *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries*. pp. 408–420.

Griffiths, T. L. and M. Steyvers (2004), ‘Finding scientific topics’. *Proceedings of the National Academy of Sciences* **101**(Suppl. 1), 5228–5235.

Hamilton, J. D. (1994), *Time series analysis*, Vol. 2. Princeton university press Princeton.

Hauff, C. and L. Azzopardi (2005), ‘Age Dependent Document Priors in Link Structure Analysis’. In: *Proceedings of the 27th European conference on Advances in Information Retrieval*. pp. 552–554.

Hauff, C., L. Azzopardi, and D. Hiemstra (2009), ‘The Combination and Evaluation of Query Performance Prediction Methods’. In: *Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval*, Vol. 5478 of *ECIR ’09*. pp. 301–312.

Hauff, C., L. Azzopardi, D. Hiemstra, and F. de Jong (2010), ‘Query Performance Prediction: Evaluation Contrasted with Effectiveness’. In: *Proceedings of the 32nd European Conference on IR Research on Advances in Information Retrieval*. pp. 204–216.

Hauff, C., D. Hiemstra, and F. de Jong (2008), ‘A Survey of Pre-Retrieval Query Performance Predictors’. In: *Proceedings of the 17th ACM conference on Information and knowledge management*. pp. 1419–1420.

He, J. and T. Suel (2011), ‘Faster Temporal Range Queries over Versioned Text’. In: *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 565–574.

He, J., H. Yan, and T. Suel (2009), 'Compact Full-text Indexing of Versioned Document Collections'. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. pp. 415–424.

He, Q., K. Chang, and E.-P. Lim (2007), 'Analyzing Feature Trajectories for Event Detection'. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 207–214.

Herscovici, M., R. Lempel, and S. Yogev (2007), 'Efficient Indexing of Versioned Document Sequences'. In: *Proceeding of the 29th European Conference on IR Research on Advances in Information Retrieval*. pp. 76–87.

Hiemstra, D. (2002), 'Term-specific Smoothing for the Language Modeling Approach to Information Retrieval: The Importance of a Query Term'. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 35–41.

Hoffart, J., F. M. Suchanek, K. Berberich, and G. Weikum (2013), 'YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia'. *Artif. Intell.* **194**, 28–61.

Holt, C. C. (2004), 'Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages'. *International Journal of Forecasting* **20**(1), 5–10.

Holzmann, H., N. Tahmasebi, and T. Risse (2013), 'BlogNEER: Applying Named Entity Evolution Recognition on the Blogosphere?'. In: *Proceedings of the 3rd International Workshop on Semantic Digital Archives*. pp. 28–39.

Jatowt, A., É. Antoine, Y. Kawai, and T. Akiyama (2015), 'Mapping Temporal Horizons: Analysis of Collective Future and Past related Attention in Twitter'. In: *Proceedings of the 24th International Conference on World Wide Web*. pp. 484–494.

Jatowt, A. and C.-m. 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*. pp. 1259–1264.

Jatowt, A., C.-M. Au Yeung, and K. Tanaka (2013), 'Estimating Document Focus Time'. In: *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*. pp. 2273–2278.

Jatowt, A., Y. Kawai, and K. Tanaka (2005), 'Temporal Ranking of Search Engine Results'. In: *Proceedings of the 6th International Conference on Web Information Systems Engineering*. pp. 43–52.

Jatowt, A., Y. Kawai, and K. Tanaka (2007), 'Detecting Age of Page Content'. In: *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management*. pp. 137–144.

Jatowt, A. and K. Tanaka (2012), 'Large Scale Analysis of Changes in English Vocabulary over Recent Time'. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 2523–2526.

Joho, H., A. Jatowt, and R. Blanco (2014a), 'NTCIR Temporalia: a Test Collection for Temporal Information Access Research'. In: *Proceedings of the 23rd International Conference on World Wide Web (Companion)*. pp. 845–850.

Joho, H., A. Jatowt, R. Blanco, H. Naka, and S. Yamamoto (2014b), 'Overview of NTCIR-11 Temporal Information Access (Temporalia) Task'. In: *Proceedings of the NTCIR-11 Conference, Tokyo, Japan*.

Joho, H., A. Jatowt, and B. Roi (2013), 'A Survey of Temporal Web Search Experience'. In: *Proceedings of the 22nd International Conference on World Wide Web (Companion)*. pp. 1101–1108.

Jones, R. and F. Diaz (2007), 'Temporal Profiles of Queries'. *ACM Trans. Inf. Syst.* **25**.

Kalczynski, P. J. and A. Chou (2005), 'Temporal Document Retrieval Model for Business News Archives'. *Inf. Process. Manage.* **41**, 635–650.

Kaluarachchi, A. C., A. S. Varde, S. Bedathur, G. Weikum, J. Peng, and A. Feldman (2010a), 'Incorporating terminology evolution for query translation in text retrieval with association rules'. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. pp. 1789–1792.

Kaluarachchi, A. C., A. S. Varde, J. Peng, and A. Feldman (2010b), 'Intelligent Time-Aware Query Translation for Text Sources'. In: *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

Kanhabua, N., R. Blanco, and M. Matthews (2011), 'Ranking Related News Predictions'. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. pp. 755–764.

Kanhabua, N. and W. Nejdl (2013), 'Understanding the Diversity of Tweets in the Time of Outbreaks'. In: *Proceedings of the 22nd International Conference on World Wide Web (Companion Volume)*. pp. 1335–1342.

Kanhabua, N. and W. Nejdl (2014), 'On the Value of Temporal Anchor Texts in Wikipedia'. In: *SIGIR 2014 Workshop on Temporal, Social and Spatially-aware Information Access (TAIA'2014)*.

Kanhabua, N., T. N. Nguyen, and W. Nejdl (2015), 'Learning to Detect Event-Related Queries for Web Search'. In: *Proceedings of the 24th International Conference on World Wide Web Companion - Companion Volume*. pp. 1339–1344.

Kanhabua, N., T. N. Nguyen, and C. Niederée (2014), 'What Triggers Human Remembering of Events?: A Large-scale Analysis of Catalysts for Collective Memory in Wikipedia'. In: *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*. pp. 341–350.

Kanhabua, N. and K. Nørvåg (2008), 'Improving Temporal Language Models for Determining Time of Non-timestamped Documents'. In: *Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*. pp. 358–370.

Kanhabua, N. and K. Nørvåg (2010a), 'Determining Time of Queries for Re-ranking Search Results'. In: *Proceedings of the 14th European conference on Research and advanced technology for digital libraries*. pp. 261–272.

Kanhabua, N. and K. Nørvåg (2010b), 'Exploiting Time-based Synonyms in Searching Document Archives'. In: *Proceedings of the 10th ACM/IEEE Joint Conference on Digital Libraries*. pp. 79–88.

Kanhabua, N. and K. Nørvåg (2010c), 'QUEST: Query Expansion using Synonyms over Time'. In: *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*. pp. 595–598.

Kanhabua, N. and K. Nørvåg (2011), 'Time-based Query Performance Predictors'. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. pp. 1181–1182.

Kanhabua, N. and K. Nørvåg (2012), 'Learning to Rank Search Results for Time-sensitive Queries'. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 2463–2466.

Kanhabua, N., S. Romano, and A. Stewart (2012), 'Identifying Relevant Temporal Expressions for Real-World Events'. In: *SIGIR 2012 Workshop on Time-aware Information Access (TAIA'2012)*.

Ke, Y., L. Deng, W. Ng, and D.-L. Lee (2006), 'Web Dynamics and Their Ramifications for the Development of Web Search Engines'. *Computer Networks* **50**(10), 1430–1447.

Keikha, M., S. Gerani, and F. Crestani (2011a), 'TEMPER: A Temporal Relevance Feedback Method'. In: *Proceedings of the 33rd European Conference on IR Research on Advances in Information Retrieval*. pp. 436–447.

Keikha, M., S. Gerani, and F. Crestani (2011b), 'Time-based Relevance Models'. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. pp. 1087–1088.

Kleinberg, J. (2003), 'Bursty and Hierarchical Structure in Streams'. *Data Min. Knowl. Discov.* **7**, 373–397.

Koen, D. B. and W. Bender (2000), 'Time Frames: Temporal Augmentation of the News'. *IBM Systems Journal* **39**(3&4), 597–616.

Kotsakos, D., T. Lappas, D. Kotzias, D. Gunopulos, N. Kanhabua, and K. Nørvåg (2014), 'A Burstiness-aware Approach for Document Dating'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. pp. 1003–1006.

Kraaij, W. (2005), 'Variations on Language Modeling for Information Retrieval'. *SIGIR Forum* **39**(1), 61.

Kulkarni, A., J. Teevan, K. M. Svore, and S. T. Dumais (2011), 'Understanding Temporal Query Dynamics'. In: *Proceedings of the Forth International Conference on Web Search and Web Data Mining*. pp. 167–176.

Kullback, S. and R. A. Leibler (1951), 'On Information and Sufficiency'. *Ann. Math. Statist.* **22**(1), 79–86.

Kumar, A., M. Lease, and J. Baldridge (2011), 'Supervised Language Modeling for Temporal Resolution of Texts'. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. pp. 2069–2072.

Lampos, V. and N. Cristianini (2012), 'Nowcasting Events from the Social Web with Statistical Learning'. *ACM Trans. Intell. Syst. Technol.* **3**(4), 72:1–72:22.

Lavrenko, V. and W. B. Croft (2001), 'Relevance based language models'. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 120–127.

Lester, N., A. Moffat, and J. Zobel (2008), 'Efficient Online Index Construction for Text Databases'. *ACM Trans. Database Syst.* **33**(3), 19:1–19:33.

Li, C. and A. Sun (2014), 'Fine-grained Location Extraction from Tweets with Temporal Awareness'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 43–52.

Li, C., Y. Wang, P. Resnick, and Q. Mei (2014), 'ReQ-ReC: High Recall Retrieval with Query Pooling and Interactive Classification'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. pp. 163–172.

Li, X. and W. B. Croft (2003), 'Time-based Language mMdels'. In: *Proceedings of the 12th international conference on Information and knowledge management*. pp. 469–475.

Llidó, D., R. B. Llavori, and M. J. A. Cabo (2001), 'Extracting Temporal References to Assign Document Event-Time Periods'. In: *Proceedings of the 12th International Conference on Database and Expert Systems Applications*. pp. 62–71.

Lumezanu, C., N. Feamster, and H. Klein (2012), '#bias: Measuring the Tweeting Behavior of Propagandists'. In: *Proceedings of the Sixth International Conference on Weblogs and Social Media*.

Mani, I. and G. Wilson (2000), 'Robust Temporal Processing of News'. In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. pp. 69–76.

Manning, C. D., P. Raghavan, and H. Schütze (2008), *Introduction to Information Retrieval*. Cambridge University Press.

Matthews, M., P. Tolchinsky, R. Blanco, J. Atserias, P. Mika, and H. Zaragoza (2010), 'Searching Through Time in the New York Times'. In: *HCIR Workshop on Bridging Human-Computer Interaction and Information Retrieval*.

Mazeika, A., T. Tylenda, and G. Weikum (2011), 'Entity Timelines: Visual Analytics and Named Entity Evolution'. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pp. 2585–2588.

McCreadie, R., C. Macdonald, and I. Ounis (2014), 'Incremental Update Summarization: Adaptive Sentence Selection Based on Prevalence and Novelty'. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. pp. 301–310.

Meeter, M., J. M. J. Murre, and S. M. J. Janssen (2005), 'Remembering the News: Modeling Retention Data from a Study with 14,000 Participants'. *Memory & Cognition* **33**(5), 793–810.

Metzler, D., R. Jones, F. Peng, and R. Zhang (2009), 'Improving Search Relevance for Implicitly Temporal Queries'. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. pp. 700–701.

Mishra, N., R. W. White, S. Ieong, and E. Horvitz (2014), 'Time-critical Search'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 747–756.

Nguyen, T. N. and N. Kanhabua (2014), 'Leveraging Dynamic Query Subtopics for Time-aware Search Result Diversification'. In: *Proceedings of the 36th European Conference on Advances in Information Retrieval*. pp. 222–234.

Niederée, C., N. Kanhabua, F. Gallo, and R. H. Logie (2015), 'Forgetful Digital Memory: Towards Brain-Inspired Long-Term Data and Information Management'. *(To appear) SIGMOD Record*.

Nørvåg, K. (2004), 'Supporting Temporal Text-Containment Queries in Temporal Document Databases'. *Journal of Data & Knowledge Engineering* **49**(1), 105–125.

Nørvåg, K. and A. O. Nybø (2005), 'Improving Space-Efficiency in Temporal Text-Indexing'. In: *Proceedings of 10th International Conference on Database Systems for Advanced Applications*. pp. 791–802.

Nørvåg, K. and A. O. Nybø (2006), 'DyST: Dynamic and Scalable Temporal Text Indexing'. In: *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning*. pp. 204–211.

Ntoulas, A., J. Cho, and C. Olston (2004), 'What's New on the Web?: The Evolution of the Web from a Search Engine Perspective'. In: *Proceedings of the 13th International Conference on World Wide Web*. pp. 1–12.

Nunes, S., C. Ribeiro, and G. David (2007), 'Using Neighbors to Date Web Documents'. In: *Proceedings of the 9th annual ACM international workshop on Web information and data management*. pp. 129–136.

Nunes, S., C. Ribeiro, and G. David (2008), 'Use of Temporal Expressions in Web Search'. In: *Proceedings of the 30th European Conference on IR Research on Advances in Information Retrieval*. pp. 580–584.

Odijk, D., G. Santucci, M. d. Rijke, M. Angelini, and G. Granato (2012), 'Time-Aware Exploratory Search: Exploring Word Meaning through Time'. In: *SIGIR 2012 Workshop on Time-aware Information Access*.

Olston, C. and M. Najork (2010), 'Web Crawling'. *Found. Trends Inf. Retr.* **4**(3), 175–246.

Parikh, N. and N. Sundaresan (2008), 'Scalable and Near Real-time Burst Detection from eCommerce Queries'. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 972–980.

Peetz, M.-H. and M. de Rijke (2013), 'Cognitive Temporal Document Priors'. In: *Proceedings of the 35th European conference on Advances in Information Retrieval*. pp. 318–330.

Peetz, M.-H., E. Meij, and M. Rijke (2014), 'Using Temporal Bursts for Query Modeling'. *Information Retrieval* **17**(1), 74–108.

Pentzold, C. (2009), 'Fixing the Floating Gap: The Online Encyclopedia Wikipedia as a Global Memory Place'. *Memory Studies* **2**(2), 255–272.

Perkiö, J., W. Buntine, and H. Tirri (2005), 'A temporally adaptive content-based relevance ranking algorithm'. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 647–648.

Pickens, J., M. Cooper, and G. Golovchinsky (2010), 'Reverted Indexing for Feedback and Expansion'. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 1049–1058.

Ponte, J. M. and W. B. Croft (1998), 'A Language Modeling Approach to Information Retrieval'. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 275–281.

Radinsky, K., E. Agichtein, E. Gabrilovich, and S. Markovitch (2011), 'A Word at a Time: Computing Word Relatedness Using Temporal Semantic Analysis'. In: *Proceedings of the 20th International Conference on World Wide Web*. pp. 337–346.

Radinsky, K. and E. Horvitz (2013), 'Mining the Web to Predict Future Events'. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. pp. 255–264.

Radinsky, K., K. Svore, S. Dumais, J. Teevan, A. Bocharov, and E. Horvitz (2012), 'Modeling and Predicting Behavioral Dynamics on the Web'. In: *Proceedings of the 21st international conference on World Wide Web*. pp. 599–608.

Rayson, P., D. Archer, and N. Smith (2005), 'VARD versus WORD: A Comparison of the UCREL Variant Detector and Modern Spellcheckers on English Historical Corpora'. In: *Corpus Linguistics 2005*.

Richardson, M. (2008), 'Learning About the World Through Long-term Query Logs'. *ACM Trans. Web* **2**(4), 21:1–21:27.

Risvik, K. M. and R. Michelsen (2002), 'Search engines and Web dynamics'. *Computer Networks* **39**(3), 289–302.

Rybak, J., K. Balog, and K. Nørvåg (2014), 'Temporal Expertise Profiling'. In: *Proceedings of the 36th European Conference on IR Research*. pp. 540–546.

Sakaki, T., M. Okazaki, and Y. Matsuo (2010), 'Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors'. In: *Proceedings of the 19th international conference on World wide web*. pp. 851–860.

SalahEldeen, H. M. and M. L. Nelson (2013), 'Carbon dating the web: estimating the age of web resources'. In: *Proceedings of the 22nd international conference on World Wide Web (Companion)*. pp. 1075–1082.

Sang, E. T. K. and J. Bos (2012), 'Predicting the 2011 Dutch Senate Election Results with Twitter'. In: *Proceedings of the Workshop on Semantic Analysis in Social Media*. pp. 53–60.

Schilder, F. and C. Habel (2003), 'Temporal Information Extraction for Temporal Question Answering'. In: *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium*. pp. 35–44.

Shan, D., W. X. Zhao, R. Chen, B. Shu, Z. Wang, J. Yao, H. Yan, and X. Li (2012), 'EventSearch: A System for Event Discovery and Retrieval on Multi-type Historical Data'. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1564–1567.

Shaparenko, B., R. Caruana, J. Gehrke, and T. Joachims (2005), 'Identifying Temporal Patterns and Key Players in Document Collections'. In: *Proceedings of IEEE ICDM Workshop on Temporal Data Mining: Algorithms, Theory and Applications*. pp. 165–174.

Shokouhi, M. (2011), 'Detecting Seasonal Queries by Time-series Analysis'. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 1171–1172.

Shokouhi, M. and K. Radinsky (2012), 'Time-sensitive Query Auto-completion'. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 601–610.

Sipos, R., A. Swaminathan, P. Shivaswamy, and T. Joachims (2012), 'Temporal Corpus Summarization Using Submodular Word Coverage'. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. pp. 754–763.

Song, W., Y. Zhang, H. Gao, T. Liu, and S. Li (2011), 'HITSCIR System in NTCIR-9 Subtopic Mining Task'.

Spina, D., J. Gonzalo, and E. Amigó (2014), 'Learning Similarity Functions for Topic Detection in Online Reputation Monitoring'. In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. pp. 527–536.

Stefanidis, K., E. Ntoutsi, M. Petropoulos, K. Nørvåg, and H.-P. Kriegel (2013), 'A Framework for Modeling, Computing and Presenting Time-Aware Recommendations'. *T. Large-Scale Data- and Knowledge-Centered Systems* **10**, 146–172.

Strötgen, J., O. Alonso, and M. Gertz (2012), 'Identification of Top Relevant Temporal Expressions in Documents'. In: *Proceedings of the 2nd Temporal Web Analytics Workshop*. pp. 33–40.

Strötgen, J. and M. Gertz (2010), 'HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. pp. 321–324.

Strötgen, J. and M. Gertz (2012), 'Event-centric Search and Exploration in Document Collections'. In: *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*. pp. 223–232.

Styskin, A., F. Romanenko, F. Vorobyev, and P. Serdyukov (2011), 'Recency Ranking by Diversification of Result Set'. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pp. 1949–1952.

Svore, K. M., J. Teevan, S. T. Dumais, and A. Kulkarni (2012), 'Creating Temporally Dynamic Web Search Snippets'. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 1045–1046.

Tahmasebi, N., G. Gossen, N. Kanhabua, H. Holzmann, and T. Risse (2012), 'NEER: An Unsupervised Method for Named Entity Evolution Recognition'. In: *Proceedings the 24th International Conference on Computational Linguistics*. pp. 2553–2568, ACL.

Tahmasebi, N., T. Iofciu, T. Risse, C. Niedereé, and W. Siberski (2008), 'Terminology Evolution in Web Archiving: Open Issues'. In: *Proceedings of the 8th IWAW*.

Tilahun, G., A. Feuerverger, and M. Gervers (2012), 'Dating Medieval English Charters'. *The Annals of Applied Statistics* **6**(4), 1615–1640.

Tran, G. B., T. Tran, N.-K. Tran, M. Alrifai, and N. Kanhabua (2013), 'Leveraging Learning To Rank in an Optimization Framework for Timeline Summarization'. In: *SIGIR 2013 Workshop on Time-aware Information Access (TAIA'2013)*.

Tran, N. K., A. Ceroni, N. Kanhabua, and C. Niederée (2015), '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*. pp. 339–348.

UzZaman, N., R. Blanco, and M. Matthews (2012), 'TwitterPaul: Extracting and Aggregating Twitter Predictions'. *CoRR* **abs/1211.6496**.

Verhagen, M., I. Mani, R. Sauri, J. Littman, R. Knippen, S. B. Jang, A. Rumshisky, J. Phillips, and J. Pustejovsky (2005), 'Automating Temporal Annotation with TARSQI'. In: *Proceedings of ACL'2005*.

Vlachos, M., C. Meek, Z. Vagena, and D. Gunopulos (2004), 'Identifying Similarities, Periodicities and Bursts for Online Search Queries'. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. pp. 131–142.

White, R. W., P. N. Bennett, and S. T. Dumais (2010), 'Predicting Short-term Interests Using Activity-based Search Context'. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. pp. 1009–1018.

Whiting, S. and J. M. Jose (2014), 'Recent and Robust Query Auto-completion'. In: *Proceedings of the 23rd International Conference on World Wide Web*. pp. 971–982.

Whiting, S., K. Zhou, J. Jose, and M. Lalmas (2013), 'Temporal Variance of Intents in Multi-faceted Event-driven Information Needs'. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 989–992.

Yan, R., X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang (2011), 'Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution'. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 745–754.

Yu, P. S., X. Li, and B. Liu (2004), 'On the Temporal Dimension of Search'. In: *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. pp. 448–449.

Zhai, C. and J. D. Lafferty (2004), 'A study of smoothing methods for language models applied to information retrieval'. *ACM Trans. Inf. Syst.* **22**(2), 179–214.

Zhang, J. and T. Suel (2007), 'Efficient Search in Large Textual Collections with Redundancy'. In: *Proceedings of the 16th International Conference on World Wide Web*. pp. 411–420.

Zhang, R., Y. Konda, A. Dong, P. Kolari, Y. Chang, and Z. Zheng (2010), 'Learning Recurrent Event Queries for Web Search'. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. pp. 1129–1139.

Zhao, X. W., Y. Guo, R. Yan, Y. He, and X. Li (2013), 'Timeline Generation with Social Attention'. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval.* pp. 1061–1064.

Zhou, K., S. Whiting, J. M. Jose, and M. Lalmas (2013), 'The Impact of Temporal Intent Variability on Diversity Evaluation'. In: *Proceedings of the 35th European Conference on Advances in Information Retrieval.* pp. 820–823.