Handling Temporal Information in Web Search Engines

Edimar Manica
Campus Avançado Ibirubá, IFRS
Ibirubá, Brazil
II, UFRGS
Porto Alegre, Brazil
emanica@inf.ufrgs.br

Carina F. Dorneles
INE/CTC, UFSC
Florianópolis, Brazil
dorneles@inf.ufsc.br

Renata Galante
II, UFRGS
Porto Alegre, Brazil
galante@inf.ufrgs.br

ABSTRACT

The Web can be considered a vast repository of temporal information, as it daily receives a huge amount of new pages. Generally, users are interested in information related to a specific temporal interval. In the information retrieval area, researches have newly incorporated the temporal dimension to the search engines. This paper presents a comprehensive study that describes the evolution of search engines on the exploitation of temporal information. Research directions and future perspectives are also presented, considering the authors' point of view.

Keywords

Temporal information, temporal search engine, keyword search

1. INTRODUCTION

Web pages describe several topics, such as conferences, sports, politics and entertainment. The most of those events change over time. The SIGMOD conference, for instance, occurs every year. The World Cup occurs every four years. Dr. House series is a television medical drama displayed once a week. The database community has devoted extensive amount effort for indexing and querying temporal data in past decades [23, 13]. In recent years, the use of temporal expressions has emerged in Web search queries once Web documents also have temporal information, as we presented in previous work [14]. However, insufficient amount attention has been given to temporal queries on the Web, bringing a new challenge of the query processing on the Web: to take into account the temporal interval desired by the user.

Another important requirement for Web queries is that people are interested in latest information, e.g., "`Who are the last FIFA World Cup champions?`". Day after day, a huge number of new pages are posted on the Web. Most of these pages are available for a long time remaining a large repository of historical information. Moreover, users also can be interested in past (e.g., "`Which are the SIGMOD articles published in 2009?`" and "`What team won the World Cup in 2002?`") or future data (e.g., "`What is the weather forecast for next Monday?`" and "`What will happen in the Dr. House series in the next chapter of Tuesday?`").

The time concept can either help in recreating a particular historical period or describing the context of a document or collection. Furthermore, the time may be useful to improve the methods of ranking results by relevance [2]. The information retrieval area adopts this new insight by adding time dimension on ranking the results. The first initiative was to sort the results by considering the time, so that the latest results are shown on the top of the ranking through the concept of *freshness metric*. Here, we classify these search engines as r t Ge er to . In the following, the concept of "filter results" is introduced in order to allow the user interaction by defining the notion of *temporal window of interest*. These search engines are here classified as Seco d Ge er to . Finally, search engines have begun to exploit the temporal information present in the Web documents content incorporating the *content-based temporal retrieval*. They differ from the previous generations since the first ones use only temporal information stored in Web documents metadata. We classify these search engines as rd Ge er to .

This paper presents a comprehensive study that describes the evolution of search engines on the exploitation of temporal information. We first discuss some special concepts used as base of our proposal. Then, we propose a new way for clustering search engines according their evolution features, categorizing them into three distinct generations, as already discussed before. A number of interesting research directions and future perspectives are also presented, considering the authors' point of view.

The rest of this paper is organized as follows. Section 2 discusses basic concepts used as groundwork in our categorization. Section 3 presents the study we have done in order to describe the evolution of search engines on the exploitation of temporal information and to propose the three generations. Section 4 points a number of interesting research directions that are open in relation to use temporal information in search engines.

2. BASIC CONCEPTS

A Web page can have different temporal information resources, which have been classified in three types [16], as described below.

- o te t - a Web document's content can contain words and/or expressions with temporal meaning (e.g. "today", "2011/10/16", "Christmas")[1].
- R ddre - all public Web documents have at least one unique address. The different segments of an URL, namely host, path and search part, might be used as a source of temporal information. For instance, a current New York Times URL is structured in the following way - http: //www.nytimes.com /2011 /01 /26 /us/ politics/26speech.html?_r=1&hp. It is possible to derive the document's year, month and day of publication by parsing this URL.
- rotocol - the Hypertext Transfer Protocol (HTTP) is an application level protocol used to request and transmit hypertext documents and components between user applications and online servers. According to HTTP protocol, servers reply to each request, sending standard headers and, if available, the requested resource (body of the message). HTTP headers have a field that represents the date and time at which the resource was last modified (Last-Modified field). However, this field is not always available and may not return a valid date, mainly, due to incorrectly configured Web servers, or because the Last-Modified field indicates the date and time at which the origin server "believes" the variant was last modified[2].

Other temporal information that can be associated to a Web page by a search engine is the cr led d te. Crawled date represents the date in which a Web page was indexed by the search engine. The temporal information presented in Web documents, such as a point in the time, an event or a period in the time, can be described in a conceptual level by a tempor l e t t . A sequence of tokens that represents an instance of a temporal entity is denominated tempor l e pre o . Those temporal concepts have been introduced in [2] together with three temporal expression categories:

- pl c t - temporal expressions that directly describe an input in a timeline[3], such as an exact date or a specific year. For example, the expressions "December, 2004" or "January 12, 2006" in a text fragment are explicit temporal expressions and can be mapped directly into points in a timeline.
- Impl c t - temporal expressions that need predefined knowledge (ontology describing temporal information, for instance) in order to be mapped into an input in a timeline. Holiday names and specific events are typical examples of implicit temporal expressions. For instance, the expression "2005 Christmas" needs to be mapped to "December 25, 2005".
- Rel t e - temporal expressions that represent temporal entities that can be only mapped into an input in a timeline in reference to an explicit or implicit temporal expression, or in reference to the moment the text has been written. For instance, the expression "yesterday" can only be mapped into a point in the time if we know the moment the document has been created.

The process of mapping temporal expressions into a standard format (so that it is possible to represent a point of a timeline in a standard way) is denominated tempor l e pre o orm l to . The search engines, which provide support to temporal information, index the normalized temporal expressions in order to allow a fast and consistent access to the temporal data.

On considering temporal expression in Web searches, an important concept highlights: tempo r l oper tor. A temporal operator defines temporal relations between two instants, two intervals or between an instant and an interval (*before*, *after* and *during*, for example). The temporal operators can modify the meaning of a temporal expression. For

[1] In this paper, the examples that represent a complete date are described in YYYY/MM/DD format.

[2] http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

[3] A timeline, also known as a chronology, is a linear representation of events in the order in which they occurred [2].

example, the following query "2004 elections" refers to the elections held in 2004. On the other hand, the query "elections after 2004" refers to the elections held after 2004, i.e., the temporal window of interest begins in 2005 and ends in the current year. This feature requires the appropriate temporal treatment of temporal operators. However, the current search engines manipulate temporal operators as a common keyword, i.e., they search for the occurrence of the keyword that represents the temporal operator in the indexed terms.

Allen [1] has defined thirteen temporal operators that can be used between intervals: after, before, contains, during, equal, finished by, finishes, meets, met by, overlapped by, overlaps, started by and starts. Figure 1 illustrates them. These operators are mutually exclusive, which characterize an issue that is not suitable for the context of Web searches because most queries are generic, which implies to use UNION in order to employ Allen's operators. For example, considering a query where the user wants pages that describe accidents that occurred between 2006 and the current year, and considering the Allen's operators, it would be necessary to use the operator met by together with the operator after, e.g. "accidents met by 2006 UNION accidents after 2006", to get the correct results, which probably would add a high degree of difficulty for most search engines users. Another example is a query in which a user wants pages about armed revolts that ended in 2005. Using Allen's operators, it is necessary to use the operator finishes together with the operator finished by, e.g., "armed revolts finishes 2005 UNION armed revolts finished by 2005". However, it would be more intuitive if there is a more general temporal operator specially projected to solve these cases. Probably, this generic temporal operator would be more useful than the operators finishes and finished by used separately, for example.

Figure 1: Allen's temporal operators

Manica et al. [14] have reported an analysis where it was verified that 33.96% of the queries (considering repeated queries) with temporal expressions have keywords that explicitly represent temporal operators. Considering just distinct queries, this percentage is 30.20%.

3. PAST AND PRESENT

In this section, we briefly describe an overview about temporal information management in some Web Search Engines, classifying them in three generations. Each new generation adds a new feature while keeping the old ones. The first generation of search engines introduces the concept of re e metr c. The second generation allows user interaction by defining the notion of tempor l do o tere t. Finally, third generation handles temporal information present on page content, supporting the co te t b ed tempor l retr e l. Third generation differs from the others since the previous generations use only temporal information stored in Web documents metadata.

3.1 First Generation - Freshness

The first generation of search engines is characterized by adding time while ranking the results through the re e metr c. The most recent pages are positioned at the top of the ranking[4]. The temporal information used in this case is the crawled date or the last modified date. T-Rank [4] is an algorithm which extends PageRank [17] to improve page ranking by exploring the freshness and activity of both pages and links. Traditional Search Engines, such as Google[5] and Yahoo![6] have adopted freshness metrics.

The strategy used in the first generation is to assume that most recently posted pages are most relevant to the user, since they mean to have the latest information. This assertion is typically valid for news, where the user generally wants to find novelty, since the oldest pages may have already been read (supposedly). However, this strategy does not benefit those users interested on historical information. For instance, a team X played a game on 2012/01/11 and another game on 2012/01/15. A user wants information about the first game (since he/she watched the last game, but lost the first one). Using a first generation search engines, the results at the top of the ranking refer to the pages with information about the game played on 2012/01/15. It is true once these pages have been created after the pages that have information about the game played on 2012/01/11, so they are more recent. Of course, it is possible to create a query that returns the pages about the game played on 2012/01/11 at the top of the ranking, using other metrics implemented by the search engines, as for instance, adding to the query a keyword that represents the opposing team, or the name of the stadium where the game has taken place. However, most users do

[4]Other metrics, such as page popularity, might also be considered. However, it is not the focus of this paper.
[5]http://www.google.com
[6]http://www.yahoo.com

not know how to create such a query. To solve this problem, the second generation of search engines adds the notion of tempor l do o tere t.

3.2 Second Generation - Temporal Window of Interest

The second generation of search engines defines the notion of tempor l do o tere t, where the users can specify the temporal interval that represents their interest. This temporal interval is defined in a specific field. The temporal information used in this case is the *crawled date*. Google, Chronica [7] and InfoSeek[7] are examples of second generation search engines.

Extracting the crawled date is a trivial task, since the search engines just need to obtain the current timestamp when storing the page on database. However, usually there is a gap between the page crawled date and the date to which the page content is related to. This gap mainly occurs in three situations: (i) when temporal information, present at the page content, is equal to the posting date but different to the crawled date; (ii) when the page content is about historical information; and (iii) when the page content has information about future. The following examples describe these situations.

The first situation can be described by considering, for instance, a page x posted on 2011/03/07, whose content is about games played on this date and with a few important links that point to it. Therewith, this page can be crawled on 2011/03/10 (i.e., days after the posting date). So, the search engines associate this date as the temporal information, although it is not the real temporal information of the page content.

The second situation can be exemplified by given a page posted, for instance, on 2011/12/14 with information about the FIFA 2002 World Cup and with several important links pointing to it. These features create a favorable situation for the page to be crawled on the same day it was posted. In this case, the temporal information associated to the page is 2011/12/14, although the page contains information about an event that occurred in 2002.

Finally, we describe the last situation by considering a page posted, for example, on 2011/05/20, with information about weather forecast for 2011/05/25 and with several important links pointing to it. It is another case where the page is crawled on the same day it was posted. The temporal information associated with the page is 2011/05/20 even though the page contains information about the future, i.e., 2011/05/25.

[7]http://www.infoseek.co.jp/

In order to solve the above problems, the third generation search engines focus is the exploration of temporal expressions present in the Web documents content.

3.3 Third Generation - Content-based Temporal Retrieval

The third generation of search engines exploits the temporal expressions present in the Web documents content in order to improve the search result quality supporting the co te t b ed tempor l retr e l. The search engines of this generation have two big challenges: (i) to extract the temporal expressions from Web pages content, and (ii) to define how to manipulate the temporal information. Therefore, before we describe the search engine, we present some temporal expression extraction tools, since they are a key feature of a search engine that manipulates temporal information. Section 3.3.1 presents the main tools for extracting temporal expressions while Section 3.3.2 shows some works we classified as third generation search engines.

3.3.1 Tools for Extracting Temporal Expressions

The current tools for temporal expressions extraction in Web documents use named entities extraction techniques to identify the temporal expressions. Some works propose the use of an XML document to store the annotated expressions [2]. TimeML [21] is an emerging standard for events and temporal expressions annotation. However, there are many tools for annotating temporal expressions that define their own form of annotation. Bellow, we present the main tools for temporal expressions extraction, and a brief comparison among them[8].

- ANNIE [3] is an open source extraction tool that is part of the GATE framework [6]. Besides temporal information, ANNIE extracts location, people, organization, sports information and so on. The extraction is performed from named entities. Some predefined entities are available, and it is also possible to define new ones through a rule-based language that is embedded in the tool. The output is an XML file with the annotations of the entities, identified in a specific language of the tool. The XML handling can be done through an API designed to be used with the tool (attached at the framework). ANNIE annotates explicit, implicit and relative temporal expres-

[8]It is important to notice that these tools are language dependent and most of them are specific to English Language.

Table 1: Temporal Expressions Extraction Tools

	POS	Language	Availability	Category	Normalization	Isolation
ANNIE	Yes	Own	Yes	E, I, R	No	Yes
GUTime	Yes	TimeML	Yes	E, I, R	Yes	Yes
PorTexto	No	HAREM Directives	No	E, I, R	No	Yes
Chronos	Yes	TIMEX2	No	E, R	Yes	Yes

Label: E: explicit I: implicit R: relative

sions. However, it does not perform the temporal expressions normalization.

- GUTime [10] is an open source tool designed to annotate temporal information. The GUTime requires the TreeTagger, which is a POS (*Part-of-Speech*) tagger that labels the words of a text with its morphosyntactic features, such as verb and noun. GUTime uses these labels in its temporal expression inference rules. The result of the annotation process is an XML document with the temporal expressions tagged according to the TimeML language. This tool annotates and normalizes explicit, implicit and relative temporal expressions. To normalize relative expression, such as today, tomorrow, yesterday, next month and last year, GUTime checks the local context in order to identify a reference point in time in which this temporal information are related to. Typically, the temporal reference point is the date of document publication.

- PorTexto [5] is a tool that recognizes temporal entity in Portuguese Language documents. The tool processes the document sentence by sentence, differently of other tools in which the text is processed word by word. The temporal expressions identification is performed by using expression patterns based on co-occurrences. These patterns are defined by a set of reference temporal words (PTR in Portuguese). A reference temporal word is a word that occurs in temporal expressions with at least two words. For instance, `year` is a reference temporal word because it occurs in the temporal expressions "`in last year`", "`in the year of 2009`" and "`a year ago`". If a sentence has a number, but it does not have a reference temporal word, then it is not considered a temporal expression, e.g., "`the product code is 2009`". The list of PTR is manually created. The extracted patterns are defined by regular expressions and stored in a file. It is possible to change the existing patterns and even include new ones. The output is an XML document with temporal annotations that follow the directives proposed by HAREM [19][9]. PorTexto annotates explicit, implicit and relative temporal expressions. However, it does not perform the temporal expression normalization.

- Chronos [15] is a tool designed to perform the recognition and normalization of temporal expressions. The text processing involves the extraction of tokens, a linguistic processing and the recognition of multi-words that is based on a list of 5,000 entries retrieved from WordNet[10]. After that, a set of approximately 1,000 rules is used to recognize temporal expressions and to extract information about them that are useful for the normalization process. Then, composition rules are performed to solve ambiguities when multiple labels are possible. The output is an XML document with temporal annotations in the language TIMEX2[11] [8]. Chronos annotates explicit and relative temporal expressions.

Table 1 presents a comparison among temporal annotations tools discussed in this section. In that table, we consider the following comparison items:

- `POS` - indicates whether the tool has linguistic processing.
- `Language` - indicates which language is used to annotate the temporal expressions.
- `Availability` - denotes whether the tool is available for download.
- `Category` - identifies which temporal expressions categories (E: explicit, I: implicit and R: relative) are supported by the tool, considering those described in Section 2.
- `Normalization` - indicates whether the tool normalizes the temporal expressions.

[9]HAREM is a joint assessment in the area of named entity recognition in Portuguese.
[10]http://wordnet.princeton.edu/
[11]TimeML is an extension of TIMEX2.

- `Isolation` - denotes whether each expression is singly evaluated or whether there is a process that seeks for patterns in all expressions of the page in order to disambiguate formats and gather information to improve the annotations.

Notice that the most of the tools uses some type of linguistic processing, which increases the cost of processing. Each tool uses a different language for annotating temporal expressions, although all languages are based on XML. Only GUTime and ANNIE are available for use. Most tools handle with explicit, implicit and relative temporal expressions. Only GUTime and Chronos perform the temporal expression normalization. All tools individually evaluate each temporal expression.

The growing availability of collections with annotated expressions allows the application of supervised machine learning techniques for the task of recognizing temporal expressions. ATEL [11] and Alias-i's LingPipe[12] are examples of these systems. The first one uses Support Vector Machine (SVM) while the other one uses Hidden Markov Model (HMM).

Several types of documents are available on the Web such as Web pages, XML documents, PDF documents, etc. The techniques for extracting temporal information may not be suitable for all types of documents. For example, the tools described above are not suitable for data-centric XML documents[13]. These documents rarely contain sentences with morphosyntactic elements, because generally they have only nodes with nouns. Therefore, tools that use linguistic processing are not proper for data-centric XML documents since they perform unnecessary processing. Moreover, the fact that these tools evaluate each temporal expression in isolation generates the loss of valuable information for the temporal expressions normalization. The process of grouping the terms according to the path that contains them and analyzing together all expressions of the same path provides more information to the normalization process. For example, the temporal expression format shown in line 4 in Figure 2 is ambiguous. It means that it is impossible to find out whether the temporal expression is related to "March 12, 2011" or "December 3, 2011". However, when checking the other values of the same XML path (`people/person/birth`) we can infer that the format of the path is `day/month/year`.

[12]http://alias-i.com/lingpipe

[13]XML documents are classified as data-centric when their data are structured. The name of the nodes typically represents semantic annotation.

```
01.  <people>
02.   <person>
03.     <name>XYZ</name>
04.     <birth>12/03/2011</birth>
05.   </person>
06.   <person>
07.     <name>TZY</name>
08.     <birth>25/03/2011</birth>
09.   </person>
10. </people>
```

Figure 2: Example of an XML document containing temporal data

TPI [14] is a third generation temporal search engine specifically designed for data-centric XML documents. TPI defines a temporal expressions tool that clusters the expressions according to their path in order to obtain information for the normalization process. This tool also has some heuristics that identify temporal intervals and dates that are structured in different elements.

3.3.2 Search Engines

In this section, we present some third generation search engines, and a brief comparison of their main features.

- TISE [12] indexes one temporal expression per page by selecting the temporal expression that better describes the events in the Web page. In the query, the temporal predicate is specified as an interval, which must be defined in a different field of that used for other keywords. The temporal query predicate is applied to the temporal expression indexed to the page. The temporal information is represented as an interval.
- TERN [22] indexes all the temporal expressions of a Web page. In the query, the temporal predicate is posed in the same field of the other keywords. The temporal predicate is applied to any temporal expression indexed to the page. The temporal information is represented as an instant.
- Pasca [18] proposes a temporal search engine for users who wants to find out when a particular event has occurred. It means that the user does not pose a temporal predicate in the query, but receives a temporal value as a result. Pasca indexes a temporal expression for each *temporal nugget*, creating a pseudo-document. A temporal nugget is a fragment of a sentence that notifies open domain facts associated with some entity. For example, "`Michael Jackson was`

`born on August 29, 1958.`". The temporal information is represented as an instant.

Table 2 presents a comparison among the third generation of search engines. We consider the following comparison items:

- `Predicate` - indicates whether the temporal predicate is embedded in the query or it is reported in a distinct field.
- `Label` - denotes whether the temporal information is represented as an instant or as an interval.
- `Temporal index` - indicates which temporal information is indexed.
- `Query type` - denotes the query type: (i)`Temporal Selection`, temporal predicate is used to filter the query result, and (ii) `Temporal Output`, the user wants to know in which time a certain event has happened.

Notice that TERN is the only search engine that allows the temporal predicate to be posed in the same field as the other keywords in the query. This feature requires an extra step for identifying and normalizing the query temporal expressions. However, it simplifies and accelerates the creation of the query, since it is not necessary to fill several distinct fields. TISE is the only search engine that represents the temporal information as an interval. The representation as an interval is wider since it allows, for example, in the query: "`preside Senate between 2001 and 2003`", to express that "2002" is also valid.

Each search engine indexes different temporal information. Indexing only one temporal expression per page (TISE) causes the loss of relevant temporal information. Indexing each temporal nugget associated with a temporal expression (Pasca), instead of indexing all the temporal expressions in a page (TERN), has the advantage of associating a temporal expression only with the terms that are close in the page content. Most search engines allow temporal selection queries. TISE, Pasca and TERN use techniques of extraction tools for identifying and normalizing temporal expressions in order to index the temporal data in a consistent and standardized way.

The main challenge is to treat the temporal expressions formulated in a query. This issue is new and there is no appropriated treatment for temporal operators (as discussed in the end of Section 2). Notice that the temporal predicate in TISE is posed in a fixed field as an interval. PASCA has no temporal predicate. TERN has the embedded predicate in the query, but does not address temporal operators.

4. FUTURE DIRECTIONS

A number of interesting and important research directions are opened when handling temporal information in web search engines. We point out some of them below.

1. **Temporal Information Weight** - temporal information is an important feature to be considered in the ranking of Web pages. However, there are other very relevant metrics to be used, for example, the popularity. The challenge for future research is the weight of temporal information comparing to other metrics. Furthermore, it is necessary to consider the temporal proximity. For example, in a query where the user wants information from the last 3 days, information from four days ago may be relevant. As already discussed in several work such as [9, 20], we know that in the most of the Web queries, users do not know exactly what they are looking for or they do not know how to properly express their need.
2. **Embedded Temporal Query** - most temporal search engines has an additional field where the user specifies the desired time period. However, handling temporal operators is an open field. Consider, as an example, a query where the temporal information is posed in the same field of other keywords. How to find the temporal operator? The temporal operators define temporal relationships, such as "after" and "before". These operators can be explicitly specified, e.g., "`President after 2000`"). Temporal operators also may be implicit, as for example in the query: "`military regime 64 84`". In this query, the temporal relationship can be "equal", i.e. the user wants to find pages having information about military regimes that started in 1964 and ended in 1984. On the other hand, the temporal relationship can be "intersection", i.e. the user wants to find pages on any military regime that happened between 1964 and 1984. The identification of implicit temporal operators requires the semantics and context analysis of the query.
3. **Index** - the index structure of a temporal search engine should consider the temporal dimension. A traditional search engine uses a

Table 2: Comparison among Third Generation Search Engines.

	Predicate	Label	Temporal Index	Query Type
TISE	Isolated	Interval	One Timex per page	Temporal Selection
TERN	Embedded	Instant	All timex in the page	Temporal Selection
Pasca	NA	Instant	One timex per temporal nugget	Temporal Output

Label: NA: Not Applicable Timex: Temporal Expression

traditional inverted index to store the terms. Usually, a list of postings (documents identifiers and pre-computed scores) per term is used, which is a scalable technique for Web search engines. However, adding the temporal dimension, the inverted index would contain the postings for all kind of temporal information. This problem increases when we consider multi-word queries. Unless the flexibility of multi-word query in the pre-identified interest is restricted, it requires a positional index. In this index, each word list contains postings for each occurrence of the word in each document. A new challenge is to add temporal dimension in order to optimize the index structure [23].

4. **Temporal Expressions Normalization** - although there are several proposals for identification, extraction and normalization of temporal expressions, there are still several gaps, such as format disambiguation and temporal interval identification. For example, having "`12/03/2011`" as a temporal expression, we can not guarantee that this temporal format is not ambiguous, since its format can be `day/month/year` or `month/day/year`. How to find the correct format? Another example, considering the identification of temporal intervals, is the sentence "`In 2004, Johny started as director of Tempo company, directing it until 2007`". This sentence has a temporal interval that begins in 2004 and ends in 2007. Thus, the search engine must have the knowledge that in 2005 Johny was also the director of the company Tempo. How to identify temporal intervals in Web pages?

5. **Temporal Ranking Queries** - the database community has devoted extensive efforts in order to index and query temporal data [23, 13]. However, insufficient attention has been given to queries with temporal ranking. For example, given any time instance t, the user could want to know about the top-k instances at the time t related to some score attribute. Ranking queries within a temporal interval rather than just at one time instance also is an open field.

6. **Temporal Evidence** - the Web is a highly dynamic environment, with significant updates occurring weekly. In this scenario, temporal evidence might be obtained from the temporal evolution of the content and structure from each individual document, and from the whole Web. This dynamic behavior creates another challenge that is: how can we use this temporal evidence to improve information retrieval? Nunes [16] discusses some challenges in this context.

5. CONCLUSION

Time is an important dimension for any application. Realizing the value of temporal information for information retrieval, researchers have begun to incorporate this dimension in Web search engines to improve their ranking mechanisms. The first initiative, and still used today, is to put the most recent pages in the top of the result. After that, some proposals have arisen with the idea of filtering the results, considering temporal intervals according to the page crawled date. Finally, search engines have been proposed in order to exploit the temporal information present in the contents of Web pages and/or the queries. This article has presented a set of search engine proposals and their mechanisms to incorporate temporal information treatment.

Finally, we suggested some challenges for future research, such as: (i) the importance of defining a temporal information weight to incorporate this feature in the ranking algorithm; (ii) the requirement for performing appropriate treatment of temporal operators in embedded temporal queries and; (iii) the necessity of modifying the traditional inverted index to aggregate a temporal dimension considering different temporal information resources (last-modified date, crawled date and temporal expressions presented in the Web page content).

6. ACKNOWLEDGMENTS

Work partially funded by the CNPq Research Grant (Process nr. 307992/2010-1. PQ 2010) and INCT (Process nr. 573871/2008-6).

7. REFERENCES

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[2] O. Alonso, M. Gertz, and R. A. Baeza-Yates. On the value of temporal information in information retrieval. *SIGIR Forum*, 41(2):35–41, 2007.

[3] ANNIE. Open source information extraction, 2010. <http://www.aktors.org/technologies/annie/>.

[4] K. Berberich, M. Vazirgiannis, and G. Weikum. T-rank: Time-aware authority ranking. In S. Leonardi, editor, *WAW*, volume 3243 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 2004.

[5] O. Craveiro, J. Macedo, and H. Madeira. Use of co-occurrences for temporal expressions annotation. In *SPIRE*, volume 5721 of *LNCS*, pages 156–164. Springer, 2009.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175, 2002.

[7] D. Efendioglu, C. Faschetti, and T. J. Parr. Chronica: a temporal web search engine. In *ICWE*, pages 119–120. ACM, 2006.

[8] L. Ferro, I. Mani, B. Sundheim, and G. Wilson. Tides temporal annotation guidelines - version 1.0.2, 2001. MITRE Technical Report MTR 01W0000041. McLean, Virginia: The MITRE Corporation. June 2001.

[9] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34:27–33, December 2005.

[10] GUTime. Adding timex3 tags, 2010. <http://www.timeml.org/site/tarsqi/modules/gutime/index.html>.

[11] K. Hacioglu, Y. Chen, and B. Douglas. Automatic time expression labeling for english and chinese text. In A. F. Gelbukh, editor, *CICLing*, volume 3406 of *Lecture Notes in Computer Science*, pages 548–559. Springer, 2005.

[12] P. Jin, J. Lian, X. Zhao, and S. Wan. Tise: A temporal search engine for web contents. In *IITA '08*, pages 220–224, Washington, USA, 2008. IEEE Computer Society.

[13] F. Li, K. Yi, and W. Le. Top- queries on temporal data. *VLDB J.*, 19(5):715–733, 2010.

[14] E. Manica, C. F. Dorneles, and R. Galante. Supporting temporal queries on xml keyword search engines. *JIDM*, 1(3):471–486, 2010.

[15] M. Negri and L. Marseglia. Recognition and normalization of time expressions: Itc-irst at tern 2004, 2004. Technical report, ITC-irst, Trento.

[16] S. Nunes. Exploring Temporal Evidence in Web Information Retrieval. In A. MacFarlane, L. Azzopardi, and I. Ounis, editors, *BCS IRSG Symposium Future Directions in Information Access (FDIA 2007)*, pages 44–50. BCS IRSG, BCS IRSG, August 2007.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[18] M. Pasca. Towards temporal web search. In *SAC*, pages 1117–1121. ACM, 2008.

[19] D. Santos, C. Freitas, H. G. Oliveira, and P. Carvalho. Second harem: New challenges and old wisdom. In *PROPOR*, volume 5190 of *LNCS*, pages 212–215. Springer, 2008.

[20] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 415–422, New York, NY, USA, 2004. ACM.

[21] TimeML. Markup language for temporal and event expessions, 2010. <http://www.timeml.org>.

[22] M. T. Vicente-Díez and P. Martínez. Temporal semantics extraction for improving web search. In *DEXA Workshops*, pages 69–73. IEEE Computer Society, 2009.

[23] G. Weikum et al. Longitudinal analytics on web archive data: Its about time! In *5th Biennial Conference on Innovative Data Systems Research (CIDR2011)*, 2011.