Chronologizing Web Pages for Effective Search

Sunil Kumar Kopparapu and Arijit De

TCS Innovation Lab - Mumbai
Tata Consultancy Services
Yantra Park, Subhashnagar,
Thane (West), Maharastra, INDIA

{SunilKumar.Kopparapu,Arijit6.D}@TCS.Com

Abstract

Search engines have become a part of our e-life. The simplest way to get near to the information that one is looking for is invariably fueled by a search engine. Due to large amount of on-line data, invariably there are multiple pages that satisfy the search criteria and hence ranking is inevitable. Most of the search engines today use some mechanism to rank the result pages and use this to display which search result page goes first. The ranking is based on information, meta or otherwise, that is readily available or easily derivable from the web pages. An important component that the search engine today does not exploit is the "age of the web page" because this temporal information is not available via the web page readily except probably for news type of information which comes usually with a date tag in the text. The "age of the page" dimension can be effectively used by search engines to rank the search results in a chronological order. For example, a search like "Monsoon in Mumbai" in the monsoon period might signify that the user is looking for information on the "current" monsoon situation rather than the highly ranked page, using some criteria, discussing about monsoon. Access to a chronologically ordered display of search results will find definite use. The reason search engines can not provide the chronological rank order is because of the absence of "age of the page" information. In the proposed paper we will elaborate on the need for dimension which helps in ranking web pages in chronological order. We investigate and discuss existing and new techniques based on natural language processing which can help in chronologizing web pages.

1 Introduction

Internet has replaced the television and the radio as the principal source of information and communication for businesses, individuals, media in all walks of life ever since it was commercialized about two decades ago. In the era of Web 2.0, the World Wide Web (WWW) had not only grown in size, but its contents have grown to become multimedia and dynamic. Today data created on the web is not merely text, but also images and videos recorded by digital cameras or web cams, streaming audio and videos extending from pod casts, net casts, music, music videos, chat transcripts, messages exchanged in discussion forums; opinions in blogs make up for the traffic and the content on the Internet. Today data gets created, updated and outdated with more speed than ever before and there is no sign of slowing down. The enormity of data created is so great that some researchers estimate that today's daily rate of content creation is about the same as the content created in three months a couple of years back. With a tremendous data explosion on the web, there is obviously a need to search through this vast volume of data to seek relevant information. This is where a Search Engine (SE) steps in. SE's have become a part of our e-life. SE's are ever popular tools to search and navigate through this ocean of information created. With burgeoning size of multimedia content on the WWW, SE's are increasingly applying more content based multi-media retrieval techniques to sort through all this information. The simplest way to get near the information that one is looking for is invariably fueled by a SE. The widespread use of SE can be captured by the fact that unlike a couple of years ago, today, we do not bookmark or remember the URLs[1] of any web page, we search for it as and when we are in need of it. The reason is (a) with ever changing landscape of the data on the web, we are likely to find different and hopefully more useful source of data than what was available when we accessed the same information earlier and (b) availability of a plethora of SE's which are able to get you to the information you seek.

Search Engines have a tough job at hand especially with the huge amount of data on the Internet that they have to sieve through to dig out the suitable web pages that match the query of the user. Too much data created asynchronously by different sources makes the task of SE's being able to produce a single output web page in response to a query very hard. This paves way for a search engine to give multiple web page output, albeit ranked, in response to a query.

[1] More popularly called the HTTP address

The popularity of SE's resonates from the usable high ranked results that they produce and what distinguishes one SE from another is the way the search output pages are ranked. Different SE's adopt different mechanisms to associate a score against each output web page and then use the score to rank the order in which they are displayed. One of the popular information that the SE's use is to give more importance to a genuine news sites like CNN rather than to a social networking site or blog or a web based bulletin board. There are several mechanisms adopted for SE's to rank order the web pages, most of these are directed towards trying to give the most relevant answer set in response to the query by understanding the intent of the query.

It is to be noted that as the web captures and records data, it implicitly creates a time sequenced snapshot of happenings or changes around us. This adds a new dimensionality: time reference to data on the WWW. This dimensionality has been mildly exploited by SE's in the news articles domain where there is an explicit correspondence between the news article and the date and time. This information is used by the SE's to not only display news in a chronological sequence but also allows user to seek latest or current news. But the same chronological ordering, unfortunately, has not been used to display the results by any of the well known SE's for non-news articles or articles where there is no explicit mention of the time corresponding to the article. When time information is not explicitly mentioned in an article it requires analysis of the content or information in a web page.

While content based retrieval techniques can now be used to search for multi-media content, it has yet to develop techniques that can actually look at an information source, be it text or multimedia, and determines the time-stamp of the information content. In this article, we discuss the need for chronolization of web pages to capture the transient and temporal nature of information evolving on the web and discuss how SE's can use this information to sort and rank web-pages in chronological order.

The rest of the article is organized as follows. In Section 2 we introduce the functioning of the SE's and show how they have been evolving since they first appeared. We discuss in Section 3, the need for SE rankings and discuss how SE's rank results they retrieve, discussing some well know ranking algorithms briefly. In Section 4 we discuss several mechanisms of combining SE results to come up with a rank of the ranked results. In Section 5 we discuss the temporal nature of the web and describe how one can capture aspects of time of information on the web which can be used for chronological analysis of web pages. it can not capture the full temporal dimension of information content and we summarize in Section 6.

2 Search Engine Overview

Typically SE's are web based tools for searching information on the World Wide Web. A SE provides a web interface where a short query can be entered. The retrieval system within the SE responds with a list of URLs or web pages relevant to the query in some sense. Often a system generated, relevance score is given along with each document, indicating the degree of relevance of each document to the query. The list of web pages is displayed in the decreasing order of relevance. Architecturally, a SE can be classified into two distinct components, namely, (a) storage and (b) retrieval.

The storage components not only help discover new web pages but also update previously discovered web pages on the web. They index and archive web pages discovered so that they can be presented to a user. The most important storage component is the web crawler which travels from a root page of a web page through links from it, to other pages, recursively navigating through the whole WWW, in a depth or breadth first crawl, in an effort to discover new and changed web pages. These pages are indexed and stored in an index database, where record of each document that the web crawler has crawled is maintained and updated.

User queries to a SE are accepted mostly (but not necessarily) through a web based user interface. Minimally, keywords from the query are used to search the index database, though essentially most SE's use some sort of query processing, which might also include cross language translation, query intent identification, user profiling, or collaborative filtering before the actual search against the index database.

The retrieval components of a SE return a set of web pages which the SE considers *relevant* to the user query. However, relevance is neither a Boolean, meaning a web page might be classified anywhere between relevant (1) and non relevant (0). This varying degree of relevance for pages makes it necessary rank them in descending order of their relevance. A SE's ranking algorithm evaluates a web page relevance score and ranks the web pages in decreasing order of their relevance to the query. While the frequency of the web crawling keeps the SE updated and current the use of complex algorithms and heuristics to rank the relevant web pages determine the ability of a SE to display 'desired' information. Structurally all the SE's are same and they differ (a) in the process of how frequently they crawl the web and updated their index database and (b) the ranking mechanism used to display the list of web pages.

Zhou [1] description of a SE is illustrated in Figure 1. Zhou refers to retrieval components as front-end processes and refers to storage components

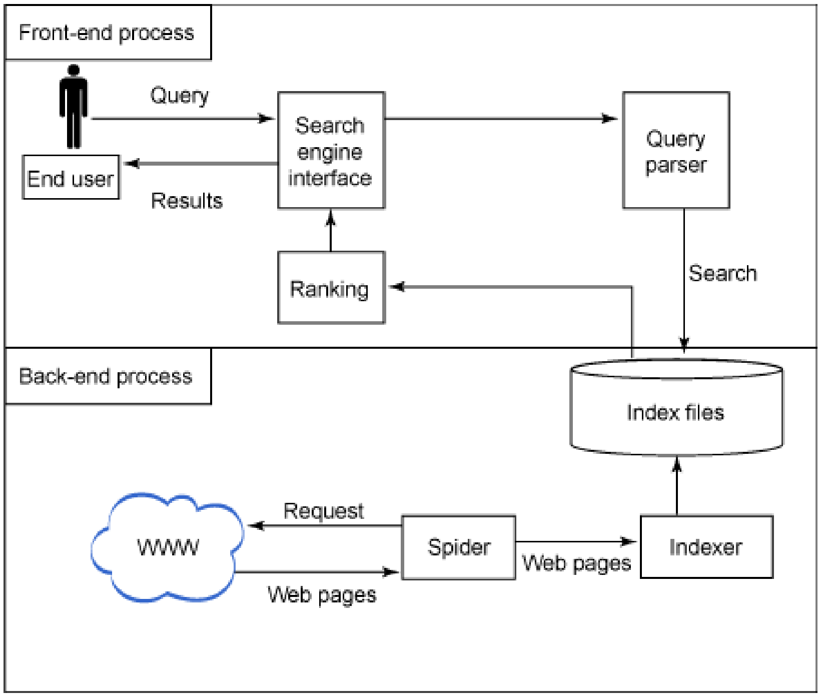

Figure 1: Typical Architecture of a Search Engine

as back-end processes.

3 Ranking Algorithms Overview

We start this section by questioning the need for ranking search engine results. When a user passes a query to a search engine, he is looking for information relevant to a query. However, what is relevant to one person, might be useless or non relevant to another. As search engines have no method of detecting query intent to a specific level, it is hard to determine what relevant information for a particular user is and what is not. Also queries to a search engine are in natural language, not in a precise, machine interpretable language. Sometimes queries are malformed and might not even give accurate information about what the user wants. There is, thus, a need for returning more than one relevant web page in a ranked order of system computed relevance, which can be accurate to some degree.

Ranking based on system computed scores for retrieved web pages is a well explored area. System Relevance scores are computed by matching query terms with words or sequence of words within a web page. Traditional measures include Term Frequency (TF), Inverse Document Frequency (IDF), TF-IDF (Term Frequency-Inverse Document Frequency) as well as other more complex measures such as Okapi BM 25 [2] and LMIR [3].

Ranking based on system computed scores for retrieved web pages is a well explored area. According to simple document ranking techniques typically view the meta tags and the text content of a web page and determine its relevance to a search query by using a variety of information such as keyword meta tags, URL information, title of the website, frequency of a query keyword, keywords in section headers, graphics, overall size of a page and proximity of keywords defines how closely correlated are the keywords.

A good example is the proximity search technique borrowed from text processing. Proximity search technique involves looking for multiple query terms that are found in a document and appear within a certain distance from each other. Sometimes proximity, searches can also involve looking for bi-grams and tri-gram sequences that match for a query and a document.

Link analysis algorithms use the structure of Internet hyperlinks pointing to a page as an effective indicator of the relevance and importance of a web page. If a web page is cited by other web pages it is considered popular. Notable link analyzing algorithms are the PageRank algorithm [4], and HITS algorithm [5]. Teoma [6] owned by Ask.com uses the HITS algorithm while Google.com [7] uses PageRank algorithm. The diagram below outlines ranking algorithms and functions.

3.1 PageRank Algorithm

PageRank extends the idea of what has been used in academic citation literature for a long time to determine the importance or quality of a page by largely counting the number of citations or back links to a given page. PageRank extends this idea by (a) not counting links from all pages equally and by (b) normalizing by the number of links on a page [4]. In case of a Topic-Based PageRank, teleporting takes place within a specific topic. So when there is a page which does not link to any other pages within a specific domain, teleporting allows a jump to another page within the same domain.

3.2 TrustRank Algorithm

Many changes in the ranking algorithm have been seen to counter spamming. The issue of web spamming to fool SE's has been one of the major test of ranking technology. Spamming pages are typically commercially driven and

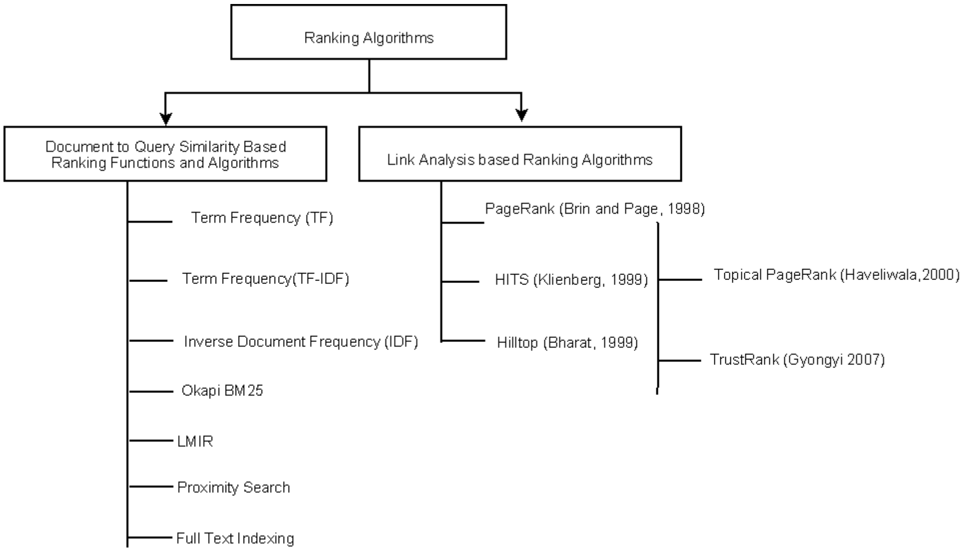
Ranking Algorithms
Document to Query Similarity Based Ranking Functions and Algorithms
Link Analysis based Ranking Algorithms
Term Frequency (TF)
Term Frequency(TF-IDF)
Inverse Document Frequency (IDF)
Okapi BM25
LMIR
Proximity Search
Full Text Indexing
PageRank (Brin and Page, 1998)
HITS (Klienberg, 1999)
Hilltop (Bharat, 1999)
Topical PageRank (Haveliwala,2000)
TrustRank (Gyongyi 2007)

Figure 2: Summary of Ranking Algorithms

aspire to boost their rankings in SE's result lists returned. Use of link farms, what provide a huge number of links to spam pages is a common approach takes by spammers. This typically beats link analysis algorithms.

Gyongyi et. al. [8] proposed TrustRank for combating spam. TrustRank is a link analysis algorithm that starts with a small set of pre evaluated expert pages. These pages are the starting points for an outward crawl to seek out similarly trustworthy pages. As the crawl moves outward, the level of trust goes down. The trust score is factored in while ranking web pages.

3.3 Hilltop algorithm

Hilltop algorithm [9] is somewhat similar to TrustRank as it employs *expert pages* and hyperlinks emanating from these pages. Expert pages are pages about specific topics that are trusted in their content. There are links from these pages to other non-affiliated pages. Some pages that are cited by a lot of expert pages are considered authorities and obtain a higher rank. The key is defining affiliation. The original algorithm states that two pages are affiliated if the first three octets of the Internet Protocol (IP) address are the same or the rightmost non generic token in the host name is the same. For example, www.bbc.com and www.bbc.co.uk would be affiliates. The most relevant expert pages and the relative distance of pages retrieved from them are used to rank the latter.

3.4 Hyperlink-Induced Topic Search (HITS)

Hyperlink-Induced Topic Search or HITS [5], is another web PageRanking algorithm that employs link analysis. The algorithm calculates two values for a page: (a) its authority, which estimates the value of the content of the page, and (b) its hub value, which estimates the value of its links to other pages. It is also known as the Hubs and Authority algorithm. The hub and authority determination is restricted to the results retrieved in response to a search query. The algorithm proceeds by recursively determine authority and hub values with an authority value being calculated as the sum of the scaled hub values that point to that page. In turn, hub values are calculated as the sum of the scaled authority values of the pages it points to. The relevance of linked pages is employed in calculating the hub and authority values.

4 Metasearching

A metasearch engine is a tool to search the web using multiple SE's in parallel. Metasearch engines have become increasingly useful with the expansion of the Web and the limited coverage of any one SE. Metasearch engines can additionally combine results from specialized SE's. A metasearch engine passes a query to multiple SE's, retrieves results from each of them in the form of a result list, and then combines these results into a single result list. Two key components of a metasearch engine are (a) the query dispatcher which decides to which SE's to send the query and (b) the result merger which decided how to merges the results from multiple SE's.

There are two types of metasearching: (a) external metasearching which involves merging results from autonomous SE's which work independent of each other. There may be an overlap in the web pages included in the search results. Typically, this type of metasearching happens when only result lists are available without any relevance score information; (b) internal metasearch, on the other hand, multiple sub SE focus on different information sources, within the same database. They return results in the form of result lists. These lists are then merged using a result merging model. In an internal metasearch engine, relevance scores are generally available from each sub SE.

4.1 Ranking Metasearch Results

Ranking of results returned by multiple SE's can be looked upon as a multi-criteria decision making problem. Each SE returns it own ranked list of web pages. The task of ranking involves merging these ranked lists into a single ranked list, thereby assigning ranks to web pages based on some combination of ranks determined by different SE's.

Result merging for metasearch is the application of data fusion techniques to search results. Data fusion techniques have been applied to develop result merging models in the past. Early research in this field includes the Logistic Regression Model [10], and the Linear Combination Model [11], [12]. Aslam and Montague proposed two models [13], the Borda-Fuse and Weighted Borda-Fuse. Diaz [14, 15] came up with a comprehensive linguistic quantifier guided fuzzy result merging model based on the Ordered Weighted Average (OWA) operator.

The Borda-Fuse model was proposed by Aslam and Montague [13] and is based on Borda-Count [16]. Each SE ranks a given set of documents. For each SE, the top ranked candidate is given d points (called Borda points). The next document receives *d-1* points and so on. The documents are ordered according to the total number of points, gained due to their position on multiple SE's. The document receiving the most points is ranked at the top. Weighted Borda Fuse model [13] determines document ranks as sum of the product of the weight attached to a particular search engine result list and the number of points accumulated by the document in that search engine results. Weights can be based on an overall assessment of the performance of the SE such as its average precision.

Diaz [14, 15] developed a fuzzy result merging model based on Yager's [17] Ordered Weighted Average (OWA) operator. The OWA operator uses a multi-criteria decision making approach where a decision function F is constructed by means of which one can combine several criteria and evaluate the degree to which an alternative satisfies the criteria. Diaz [14, 15] applies the OWA operator in result merging by determining the ranks as the degree to which the web pages (alternatives) satisfy the SE (criteria). The OWA model was further extended for metasearch by De et. al. [18]. Experimental Results [14, 15, 18] have shown improved result merging performance when using fuzzy models over Borda Fuse model.

5 Temporal Ranking of Web Pages

While most of the research in the area of search engine result ranking has been on the aspect of trying to identify the most popular or most relevant set of pages so as to enable SE's to display the results in the decreasing order of relevance. One aspect that has not been addressed is the aspect of chronological ranking of the web pages. Today SE's provide this aspect of ordering/ranking on news articles. It should be noted that there is a strong correlation between the date of publication of the news article and the contents of the news article. This correlation allows SE's to rank the articles based on the currency of the news article, thus allowing users to search for news articles which were created in the last one hour, last one week and so on. This chronological ordering or ability of a SE to determine the age of the page is possible only because there is an explicit reference to date and time in the news article. Though not explicitly mentioned in technical articles, a form of chronolization can be observed in scientific articles where one can determine the age of the article in terms of the *not published before date* by looking at the list of references (mandatory in scientific articles). The date of publication of the most recent cited reference in the list of references is the date *after* which the article has been published.

Chronolization has two aspects, (a) the date of creation or modification of the article (popularly called time stamp of the article); this is researched under the topic of Web Archiving[2] and (b) the reference in time to which the content of the article refers to. Though not mandatory, in many cases, as is evident in news articles, both the time-stamp and the reference in time of the information content might be same. But in general this is not true, for example, an article describing an incident in World War II could be written in 2009 – in this case the time stamp of the article is 2009 while the time reference of the article is 1939-45[3]. In another scenario, there can be an article which describes a scenario of the future (Sci-Fi article) written in 2009 about the nature of environment in 2020. Here the time stamp is 2009 but the information content is timed to the year 2020 (into future). Clearly there is a need to distinction between the time stamp and the time associated with the information in a page. In this article we refer to chronolization with respect to the time of the information content which is important for user searching for information.

[2] A popular web site archiving is hosted at achrieve.org

[3] 1939-1945 refers to the actual time when the World War II was fought

5.1 What is Web Page chronologizing?

Chronologization should not be confused with web archiving which takes snapshots at different times of a web page and store them in and ordered fashion based on the time of the snapshot. Chronologization is the ranking of web pages in descending order of currency of the content. If a page has information on a topic which is current then it should be ranked at the top. Chronologization helps us determine and hence rank on top the most *recent* information on a topic in response to a query and not the web page that has been most recently altered. The last updated web page on the topic might not contain the most recent information. There is a need for ranking of web pages based on chronological ordering. For example, consider a query *Monsoon in Mumbai;*. in this case, using chronological ordering, a page about the current monsoon situation in Mumbai would be ranked higher against the most popular page which refers to one of the worst monsoon in Mumbai[4]. In another example, look at a student working on a school project or a research project; (s)he would want to be able to see the most recent articles (which is likely to contain recent findings) first and the older articles (whose content or theory be obsolete) later. Ability to determine the age of the content is non-trivial and requires deep language processing, especially when there is no explicit reference to time on the page. One could in theory build a chronological WWW where each article is related to every other article in relation to time but this is hardly necessary. One could create a chronological ordering of the web pages returned by a SE (or a set of SE's) in response to a query. In this case, we need to time order a finite and a small set of web pages; which makes the chronolization process tractable and less ambiguous. Ranking based on chronolization would benefit; today none of the SE's gives an option to visualize the results based on chronological order.

5.2 How to achieve Chronolization?

Web archiving can record snapshots of pages with time stamps attached to them. Using these snapshots it is easy to track changes to the page and create a time line in which information appears and is removed from the page. However, it difficult to say if the current contents of the page contain the most recent information on the topic. It is also difficult to rank pages on the currency of information on a topic. A trivial way to do this would

[4] Today almost invariably all the SE's pop up the page corresponding to the monsoon flooding of Mumbai on July 26, 2005

be to select and rank pages in the order of their last modified times (time stamp) obtained from web archives.

Consider there are N result pages that the SE[5] returns in response to a user query. Chronolization would mean to rank the N result pages in an order that would place the page with the current information at top and rank it high while the page containing later or older information would be ranked lower. Let P_1, P_2, $\cdots$, P_N be the N pages returned by a SE in response to a query. The idea is to analyze the content of the page and determine the *PublishDate* and *ContentDate*. The *PublishDate* would be a reference to the date on which the page was published; this can be determined by using several cues like (a) using the information in archieve.org, (b) looking at the metadata for the publishing date, (c) look at the HTTP address, (d) look for a time stamp[6] in the text, (e) look for an explicit *Last Updated* like string. If there are several *PublishDate*'s, we choose the one that is closest to the current date and time[7]. Determining *ContentDate* which is useful to rank the pages in the chronological order requires language processing on the contents of the page and an access to some kind of a event-time knowledge base. A typical event-time knowledge base is shown in Table 1. One could use a minimal parsing system [19] to identify the *ContentDate* of the page. Typically, a single article could have multiple *ContentDate*'s especially if the content refers to information over a period of time[8]. All the dates associated with the *ContentDate* are meaningful and are used to rank the pages in chronological order. A simple way is to assign multiple ranks with the page having multiple *ContentDate*'s. For example, if there are two pages P_1 and P_2; let the *ContentDate* of P_1 be 15 Aug 1947 and 26 Jan 1950 and the *ContentDate* of P_2 be Dec 1949. Then we would rank the pages as P_1 (due to *ContentDate* being 26 Jan 1950), P_2, P_1 (due to *ContentDate* being 15 Aug 1947). In the rest of this section we give a heuristics based approach to order the pages in a chronological order.

Natural language processing (NLP) can be used in ranking retrieved web pages based on the information currency. Following this, NLP can be used to read through (parse) the content of web pages and segment it into sentences. The grammatical tense (present, past and future) of each sentence can be determined using corresponding rules for verb conjugation and sentence structure using NLP [20]. If a majority of the sentences in the page are written in a specific tense then we can say that the *tone of writing*

[5] could also be from multiple SE's

[6] could be in different formats

[7] system date and time

[8] a page related to history

Event	Time
World War II	1939-1945
Christmas	Dec 25
Gandhi birthday	02-10-1869
⋮	⋮
Children's Day	14 November

Table 1: Event-Time Knowledge Base

Table 2: Heuristics to determine chronological ranking of web pages

Heuristic	Tone of Web Page	Time Stamp	Chronological Ranking
1	Present	Date ≈ Current	High
2	Present	Date older than Current	Low
3	Past	Date ≈ Current	Low
4	Past	Date older than Current	Low
5	Future	Date ≈ Current	Medium
6	Future	Date older than current	Medium

of the page is in that particular tense. If the tone of the page is in present tense and the time stamp on the page is close to the current time, then it is highly likely that the page is current should be ranked high. Table 2 below simple heuristics to rank a page into three categories based on possible currency of information. A brief description of Table 2 is given below.

Heuristic 1 (Present Tense; Date approximately Current Date) This implies that most likely the web page was written recently and since it has been written in the present tense it indicates that it is describing a recent even or happening. Hence its ranking should be chronologically higher.

Heuristic 2 (Present Tense; Date older than Current Date) This implies that most likely the web page was written in the past and even though it was written in the present tense, at this time it is referring to a past even or happening. Hence its ranking should be low in the search engine list.

Heuristic 3 (Past Tense; Date approximately Current Date) This implies that most likely the web page has been written recently but recounts a past even or happening. Hence its ranking should be low in the search engine list.

Heuristic 4 (Past Tense; Date older than Current Date) This implies that the document was written in the past and recounted a historic event then. Hence its ranking should be low.

Heuristic 5 (Future Tense; Date approximately Current Date) This implies that the document was written recently but seems to recount a future even or happening. Hence it is highly likely it is reporting on a current or future event and the information content is the latest. Hence its ranking should be high

Heuristic 6 (Future Tense; Date older than Current Date) This implies that the document was written in the past but seem to recount a future even or happening. The even might have happened in the recent past or might happen in future. There is a higher likelihood of its happening in the present time. Hence the ranking should be medium.

Figure 3 and Table 2 show a scheme for chronological ranking of SE results into three categories high, medium and low based on how current the information contained in the page is. By identifying the *tone* or most prevalent tense of the textual content of the web page and determining last modified time-stamp of the web page we can rank web pages into three categories high, medium and low. When displaying results in a chronological order, web pages classified as high are displayed first, web pages classified as medium are ranked after that and web pages ranked low are classified at the bottom. Standard ranking algorithms such as PageRank can then be used to rank web pages within a category.

6 Summary

In this chapter, we have discussed the fundamental architecture of a search engine (SE). A search engine explores the web through a crawler; indexes, stores and archives web pages. A SE parses the query and in response retrieves a set of results relevant, in some way, to the query and displays them in an ranked sequence. Ranking is very crucial and the popularity of a SE rests in the way it ranks the search results. In this chapter we provided an overview of some of the most widely used ranking algorithms used by most

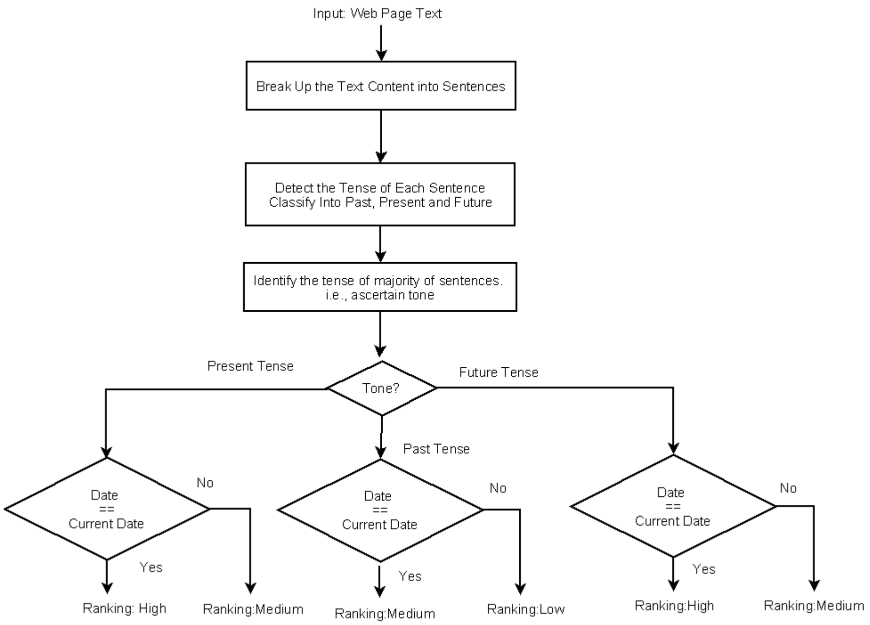

Figure 3: Ranking Web Pages by Temporal value of information content

SE's. We also discussed the temporal nature of the web and demonstrated the need for ranking results in a chronological order. Most chronological rankings are employed use time-stamp information and reference well established web archives. However, these web archiving based information, use the creation time of a page instead of the actual information content of a page. We proceed to outline a NL-based approach to determine the age of actual information content and an approach to rank web pages based on this approach.

References

[1] Deng Peng Zhou, "Beef up web search applications with lucene: Improve searches with a more robust app from the apache jakarta project," in *IBM Developer Works*, 2006.

[2] Robertson S. E., "Overview of the okapi projects," in *Journal of Documentation*, 1997, vol. 53, pp. 3–7.

[3] Zhai C. and Lafferty J. A, “Study of smoothing methods for language models applied to ad hoc information retrieval,” in *Proceedings of SIGIR*, 2001, pp. 334–342.

[4] Sergey Brin and Lawrence Page, “The anatomy of a large-scale hypertextual web search engine,” in *Computer Networks and ISDN Systems*, 1998, pp. 107–117.

[5] Jon Kleinberg, “Authoritative sources in a hyperlinked environment,” in *Journal of the ACM*, 1999, vol. 46, pp. 604–632.

[6] Chris Sherman, “Teoma vs. google, round two,” in *Search Engine Watch*, April 2002.

[7] Lawrence Page, “Method for node ranking in a linked database,” US Patent 6285999 for Google Inc, 2001.

[8] Zoltn Gyngyi, Hector Garcia-Molina, and Jan Pedersen, “Combating web spam with trustrank,” in *Proceedings of the International Conference on Very Large Data Bases*, 2004, vol. 30, p. 576.

[9] Krishna Bharat and George A. Mihaila, “Hilltop: A search engine based on expert documents,” in *WWW9 Conference*, May 15-19 2000.

[10] D.A. Hull, J. O. Pedersen, and H. Schtze, “Method combination for document filtering,” in *Proceedings of the 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, August 1996, pp. 279–287.

[11] P. Thompson, “A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model,” in *Information Processing and Management*, Nov. 1990, vol. 26, pp. 371–382.

[12] P. Thompson, “A combination of expert opinion approach to probabilistic information retrieval, part 2: mathematical treatment of ceo model,” in *Information Processing and Management*, Nov. 1990, vol. 26, pp. 383–394.

[13] J. Aslam and M. Montague, “Models for metasearch,” in *Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 276–284.

[14] A. De E. D. Diaz and V.V. Raghavan, “A comprehensive owa-based framework for result merging in metasearch,” in *Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, Sept. 2005, pp. 193–201.

[15] E. D. Diaz, *Selective Merging of Retrieval Results for Metasearch Environments*, Ph.D. thesis, University of Louisiana, Lafayette, LA, May. 2004.

[16] J. C. Borda, "Memoire sur les elections au scrutiny," in *Histoire de l'Academie Royale des Sciences*, 1781.

[17] R. R. Yager, "On ordered weighted averaging aggregation operators in multi-criteria decision making," in *Fuzzy Sets and Systems*, July 1983, vol. 10, pp. 243–260.

[18] Arijit De, Elizabeth E. Diaz, and Vijay V. Raghavan, "On fuzzy result merging for metasearch," in *FUZZ-IEEE 2007*, 23-26 July 2007, pp. 1–6.

[19] Sunil Kumar Kopparapu, Akhlesh Srivastava, and P. V. S. Rao, "Minimal parsing key concept based question answering system," in *HCI (3)*, Julie A. Jacko, Ed. 2007, vol. 4552 of *Lecture Notes in Computer Science*, pp. 104–113, Springer.

[20] Qiu Gui Su, "Determining time frames," in *http://mandarin.about.com/od/grammar/a/aspect.htm*, 2009.