Analysis of Lexical Signatures for Finding Lost or Related Documents

Seung-Taek Park[1] David M. Pennock[2] C. Lee Giles[1,2,3] Robert Krovetz[2]

[1]Department of Computer Science and Engineering
[3]School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802 USA
{separk@cse, giles@ist}.psu.edu

[2]NEC Research Institute
4 Independence Way
Princeton, NJ 08540 USA
{dpennock, krovetz}@research.nj.nec.com

ABSTRACT

A *lexical signature* of a web page is often sufficient for finding the page, even if its URL has changed. We conduct a large-scale empirical study of eight methods for generating lexical signatures, including Phelps and Wilensky's [14] original proposal (PW) and seven of our own variations. We examine their performance on the web and on a TREC data set, evaluating their ability both to uniquely identify the original document and to locate other relevant documents if the original is lost. Lexical signatures chosen to minimize document frequency (DF) are good at unique identification but poor at finding relevant documents. PW works well on the relatively small TREC data set, but acts almost identically to DF on the web, which contains billions of documents. Term-frequency-based lexical signatures (TF) are very easy to compute and often perform well, but are highly dependent on the ranking system of the search engine used. In general, TFIDF-based method and hybrid methods (which combine DF with TF or TFIDF) seem to be the most promising candidates for generating effective lexical signatures.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; E.2 [**Data**]: Data Storage Representations; H.3.7 [**Information Systems**]: Information Storage and Retrieval—*Digital Libraries*; H.3.2 [**Information Systems**]: Information Storage and Retrieval—*Information Storage*

General Terms

Algorithms, Measurement, Performance, Reliability, Experimentation, Verification.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'02, August 11-15, 2002, Tampere, Finland.
Copyright 2002 ACM 1-58113-561-0/02/0008 ...$5.00.

Keywords

lexical signatures, robust hyperlinks, dead links, broken URLs, information retrieval, World Wide Web, search engines, indexing, digital libraries, TREC, term frequency, inverse document frequency

1. INTRODUCTION

The World Wide Web is a dynamic information resource: web pages and hyperlinks are constantly being added, modified, moved, and deleted by independent entities around the world. Pitkow [15, 16] reports that around five to eight percent of requested hyperlinks on the web are broken (i.e., return an error); Lawrence et al. [12] find that many URL citations in research papers become invalid as early as a year or two after publication. Because of the web's scale, dynamics, distributed control, and lack of facilities for maintaining persistence of information, finding desired information remains a challenging problem. General-purpose search engines only cover a limited portion of the web, and most take several months to update new information [9, 10]. One solution is to build a greater variety of special-purpose search engines that can react more quickly to changes within their particular domain. For example, Lawrence, Giles, and Bollacker [11] developed ResearchIndex to collect and maintain a searchable index of computer science research papers. However, these efforts require considerable start-up and maintenance costs, and so may not be feasible for every domain.

Several initiatives address the problem of broken links by proposing mechanisms for assigning location independent names to documents in addition to URLs [3, 6, 7, 17, 18]. None of these approaches have been widely adopted because they require users either to acquire new software or to explicitly maintain the validity of name dereferencing. Other different approaches are addressed in [1, 2, 4, 5, 13]. Phelps and Wilensky [14] propose a less burdensome solution: compute a *lexical signature* for each document, or a string of about five key identifying words in the document. If the document cannot be found by URL, then it can often be located by feeding its signature words into a search engine. Phelps and Wilensky propose that lexical signature words be chosen by maximizing a modified term-frequency inverse-document-frequency (TFIDF) measure, capping term frequency (TF) at five, among other modifications. They also

propose methods for embedding lexical signatures into hyperlinks and instrumenting browsers to automatically perform content-based dereferencing (i.e., query a search engine and process the results) when standard URL dereferencing fails. Phelps and Wilensky report that, in most cases, a search engine returns the desired document and only that document. If the search engine returns no documents (because the desired document no longer exists or is not indexed), then the authors suggest removing one or more words from the lexical signature and using this reduced signature to search for substitute documents. Thus, a secondary purpose for lexical signatures is to discover relevant or similar documents when the desired document is truly lost.

We find that, because Phelps and Wilensky's method (PW) caps TF at five, it places too much emphasis on document rarity and, on huge document collections like the web, acts almost identically to the method that choose lexical signatures by minimizing DF (document frequency). Both DF and PW are good at uniquely identifying a document when it exists and is indexed by the search engine, but neither is good at finding relevant documents when the target document is not in the search engine's database. Moreover, we believe that unique identification is often unnecessary: as long as the search engine returns the desired document as the first-ranked document (even among many documents), then the lexical signature is effective.

In this paper, we study the relative efficacy of eight different methods for generating lexical signatures, including PW. We conducted tests both on actual web documents, where search engine coverage and ranking algorithms are limited, and on a TREC data set, where search coverage is complete. PW performs well on the latter data set, but more like DF on the web. Lexical signatures based on maximizing term frequency (TF) are easy to compute and maintain, since they do not depend on measuring statistics across the database. TF often performs well, but depends to a large extent on how the search engine ranks documents. TFIDF-based method and hybrid methods that use one or two minimum-DF words along with maximum-TF or maximum-TFIDF words perform well both on real web data and on the idealized TREC data, in terms of both finding the desired document and finding alternate relevant documents.

2. TERMINOLOGY

2.1 Lexical Signature

Phelps and Wilensky's [14] main motivation was to associate lexical signatures with documents, so that when the lexical signature is fed to a search engine, the desired document—and only that document—is returned. Then, when URLs change and links to documents become invalid, new locations for documents can be easily found via search engines. To achieve this goal, Phelps and Wilensky argued that lexical signatures should have following characteristics:

1. Lexical signatures should extract the desired document and only that document.
2. Lexical signatures should be robust enough to find documents that have been slightly modified.
3. New lexical signatures should have minimal overlap with existing lexical signatures.
4. Lexical signatures should have minimal search engine dependency.

We prefer a slightly weaker notion of unique identification: as long as the desired document is the top-ranked document returned by the search engine, we are satisfied. We also pay closer attention to the other potential benefit of lexical signatures: to help the user find relevant documents when the desired documents is truly lost. We therefore modify the first desired characteristic of lexical signatures as follows:

1a Lexical signatures should easily extract the desired document. When a search engine returns more than one document, the desired document should be the top-ranked document.

1b Lexical signatures should be useful enough to find relevant information when the precise documents being searched for are lost.

2.2 What is a Term?

Lexical signatures are composed of a small number of *terms*. Phelps and Wilensky [14] used individual words as terms, where words are case-insensitive, contained in the context of the document (not in meta-tags), contain at least four letters, and do not include any numbers. In our experiments, these rules of thumb for defining terms proved to be fairly effective. Number queries caused many problems with document retrieval: for example, if the query '2,000' is given to search engines, some return documents that contain only '2 000', '2;000', '2:000', or '2,000', but not '2000'. Also, many words that have less than four letters are stop words (e.g., 'the', 'of', or 'in'). In our experiments, filtering out these short words for the most part slightly improved the efficacy of the lexical signatures.

2.3 Basic and Hybrid Methods for Generating Lexical Signatures

In our experiments, we explore lexical signatures containing five terms. We generate eight kinds of lexical signatures for each document: four *basic* lexical signatures and four *hybrid* lexical signatures. Basic lexical signatures are generated using a single metric. For example, TF-based signatures are generated based on the term frequency values of words in the given document. Hybrid signatures combine terms generated from two different basic methods. For example, TF3DF2 uses three TF-based words and two DF-based words. A detailed explanation of the basic lexical signature methods follows.

Basic Lexical Signature Methods

1. **TF**: Select terms in decreasing term frequency (TF) order. If there is a tie, then pick words based on increasing document frequency (DF). If tied again, randomly select the words.
2. **DF**: Select words in increasing DF order. If there is a tie, then pick words based on decreasing TF order. If tied again, randomly select the words.
3. **TFIDF**: Select words in decreasing term-frequency inverse-document-frequency (TFIDF) order. If there is a tie, then pick words based on increasing DF order. If tied again, randomly select the words.

4. **PW**: Select words based on Phelps and Wilensky's [14] method, or decreasing TFIDF order where the TF term is capped at five. If there is a tie, then pick words based on increasing DF order. If tied again, randomly select the words.

Since we are also interested in the ability of lexical signatures to extract relevant documents when they fail to find missing documents, we find it useful to divide a lexical signature into two parts. The first part is useful for finding relevant documents and the second for uniquely identifying the desired document. If the desired document is not extracted because a search engine has not indexed it or the document is lost or gone, the second part of the lexical signature is removed and we attempt to find relevant documents using only the first part. Since we want to use the first part of lexical signatures to find relevant documents, we filter out in that part all words that have the document frequency one. This leads us to propose the following hybrid lexical signature methods:

Hybrid Lexical Signature Methods

1. **TF3DF2**: Select two words in increasing DF order. Then filter out all words which have DF value one. Select three words maximizing TF.
2. **TF4DF1**: Select one word based on increasing DF order first. Then filter out all words which have DF value one. Select four words maximizing TF.
3. **TFIDF3DF2**: Select two words based on increasing DF order first. Then filter out all words which have DF value one. Select three words maximizing TFIDF.
4. **TFIDF4DF1**: Select one word based on increasing DF order first. Then filter out all words which have DF value one. Select four words maximizing TFIDF.

2.4 Similarity Method: Cosine Measure

To measure the similarity between returned documents and the desired document, we use the cosine measure within the vector-space model [20]. For n unique words in our corpus, each document can be represented as an n-dimension vector. For example document A can be represented as

$$A = (a_1, .., a_n),$$

where if the ith word in the corpus appears in document A, then a_i is either the word's TF value, or the word's TFIDF value. Otherwise, $a_i = 0$. There is no easy way to extract the exact DF value of each word for all documents on the World Wide Web. Also, even though we could extract the DF value of each word for documents indexed by a search engine, the heavy search query burden for a search engine would not be tolerated. Thus, we use TF values with the cosine measure on actual web pages (which gives a slight bias to TF-based lexical signatures in our experiments), and use TFIDF values on TREC data (which gives a slight bias to TFIDF-based method). Using the vector space model, the cosine similarity measure between documents A and B is:

$$\cos\theta = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \sum_{i=1}^{n} b_i^2}}$$

In practice, documents on the web are regularly updated or modified. We would like to consider extremely similar documents that result from minor modifications to be the same document. Therefore, if the cosine value of two documents is greater than 0.9, we consider them to be the same document.

3. EMPIRICAL RESULTS FROM WEB DATA

Phelps and Wilensky [14] report that their lexical signatures extract only in most cases one or two documents, one of which is the desired document. However, *most cases* is not defined. What percentage of lexical signatures will extract the single desired document? If a list of documents is returned, is the desired document in that list? We postulate that if the desired document appears in the top ten of the list, especially at the first place, such lexical signatures are effective.

3.1 Web Data Set

For this experiment, we extracted the first 1500 URLs from pre-crawled data[1] (containing 1.5 million documents and their URLs) and downloaded each corresponding document. Several web documents do not have any words or have only a few words in their content, e.g. some documents only contain Java scripts or flash links. We excluded all documents and corresponding URLs that contained less than fifty words. The URLs that could not be downloaded because of server failures or corresponding document removals were also excluded. After removing stop words from word tokens, we again removed any URLs and corresponding documents that contained less than five unique words. Our original data set now reduces to 980 documents and their corresponding lexical signatures.

3.2 Experimental Method

Since there is no obvious way to get the document frequency (DF) of each word for the entire web, we used a search engine to generate a DF list for all words in our document set. In this experiment, we assumed that the DF value of each word from the search engine *Google*[2] is proportional to the actual DF value for the entire Web. Based on document frequency list and term frequency list of each document, we examine eight different lexical signatures per document.

After generating lexical signatures for all methods, we used them as queries for three search engines: *YahooGoogle*,[3] which uses *Google*'s searching algorithm, *AltaVista*,[4] and *MSN*.[5] Unlike *MSN* and *YahooGoogle*, *Altavista* returns all documents that contain any words in the lexical signature. To solve this problem, we used the advanced search option for *Altavista* (Using this option, *Altavista* only returns documents that contain all words in query). If a search engine did not return any documents, we removed a word from the

[1]The data used in this experiment was created by crawling all of dmoz.org and tnen retrieving all links external to the dmoz.org domain (i.e., all websites listed in Open Directory) .From these URLS any href tag was extracted eliminating any files with a file name extension that indicated that it was not html (e.g., pdf, ps, doc, etc). Thus the data set consists of all sites listed in DMOZ plus all sites two forward links away from DMOZ.

[2]http://www.google.com/

[3]http://google.yahoo.com/

[4]http://www.altavista.com/

[5]http://www.msn.com/

given lexical signature based on its lowest DF order and re-queried the search engine. This procedure was continued until the search engine returned documents or all of words in the given lexical signature were removed. After the search engine returned the list of documents, those documents were downloaded and the similarity between returned documents and the target document was calculated using the cosine measure. If the search engine returned more than ten documents, we analyzed only the top ten ranked documents and ignored the rest. For all queries to *YahooGoogle*, all but two queries to *MSN*, and all but four queries to *Altavista*, at least some documents were returned.

3.3 Retrieval Performance

Our concern is not only with whether or not the desired document is returned but with its location in a possible list of returned documents. We define retrieval performance of lexical signatures as the percentage of times the desired document is returned based on how and when the document is returned. Since we already have a signature for the desired document, our performance measure is similar to recall.

We define the following disjoint classes. *Unique* represents the percentage of lexical signatures that successfully extract and return the single desired document. (This is the class discussed by Phelps and Wilensky.) *Top* represents the percentage of lexical signatures that extract a list of documents with the desired document first ranked. *High* is the percentage of lexical signatures that successfully returned a list with the desired document but not first ranked, but one of top ten. *Other* represents the percentage of lexical signatures that failed to extract the desired document. Because these classes are disjoint, the above added together represent 100% of all cases. Figures 1, 2, and 3 show the retrieval performance of each lexical signatures in extracting the desired documents for three different search engines averaged over 980 unique lexical signatures.

Figures 1, 2, and 3 show that considering only the *unique* extraction property of lexical signatures is not the only important factor in extracting the desired document. Note that DF and PW are most efficient for the *unique* property. However, if we focus on just retrieving the desired document, i.e. the case where *unique*, *top*, and *high* are combined, then hybrid methods are most consistent over three different search engines and efficient. Using this definition of retrieval performance, PW and DF methods performed worse than others for *YahooGoogle*. Phelps and Wilensky argued that the original TFIDF did not sufficiently emphasize the contribution of rarity; that is, if lexical signatures are chosen to minimize DF, it would be more helpful to filter out other documents and extract only the single desired document. However, since they limit the words in the lexical signature to a TF of 5, we argue that rarity is actually overemphasized and, as the number of documents on the Web increases, the PW and DF lexical signature methods become similar which is indicated by their retrieval performances shown in Figures 1, 2, and 3.

3.4 Finding Relevant Documents

Suppose the desired document cannot be extracted; can the lexical signature find a related one? If so, which method works best? Figure 4, shows the percentage of all 980 documents not found in each search engines using all lexical signature methods; specifically 194, 172, and 232 documents

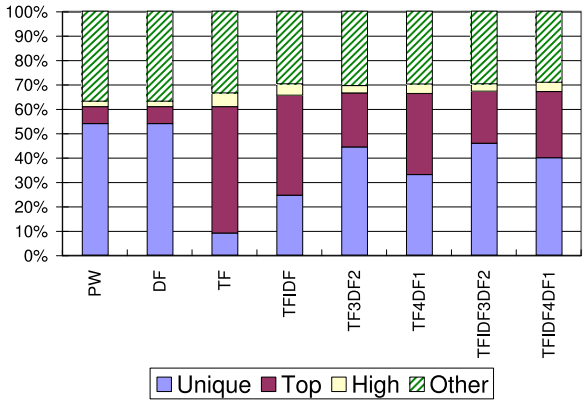


Figure 1: Retrieval performance of lexical signature methods for *YahooGoogle*

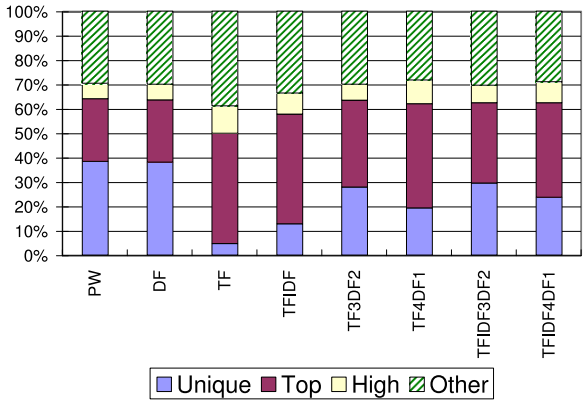


Figure 2: Retrieval performance of lexical signature methods for *MSN*

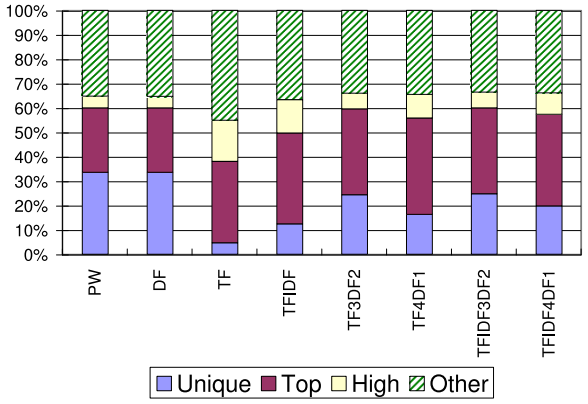
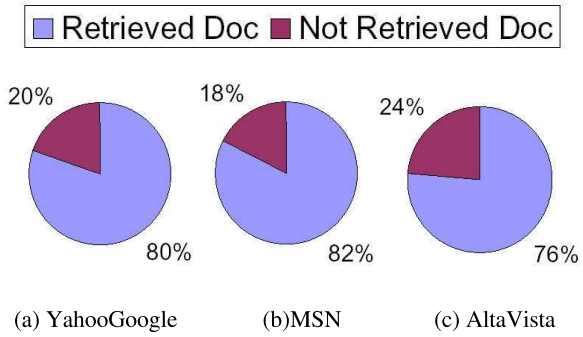


Figure 3: Retrieval performance of lexical signature methods for *AltaVista*



(a) YahooGoogle (b)MSN (c) AltaVista

Figure 4: Coverage of each search engine

(*Not Retrieved Doc*) could not be retrieved from *YahooGoogle*, *MSN*, and *Altavista*, respectively. There could be many reasons for this. The desired documents are not yet indexed by the search engines; they are moved to another server; they are deleted; they are modified or updated; they consist of very few unique words; their names are changed; the words in the lexical signature are not indexed; etc.

In those situations, lexical signatures should be expected to extract highly relevant documents. Also, if the search engine returns a list of relevant documents, then it would be useful if first ranked document be one of the most relevant ones.

We now only consider the effect of lexical signatures on retrieving related documents to the *Not Retrieved Documents* class. We analyze the cosine similarity of the documents retrieved in the *Not Retrieved Document* class to the desired document. Figure 5 gives the average cosine values of the first ranked documents and Figure 6 shows average cosine values of the top ten documents. Here, DF and PW have the worst average cosine values for both first ranked documents and top ten documents.

Figure 7 shows the average cosine values of the first ranked document for all 980 documents. In general PW, DF (because of their better retrieval performance) and hybrid methods yield better average similarity for first ranked documents than TF and TFIDF, and TF and TFIDF show more variation between search engines.

3.5 Observations

Two characteristics which are important for a successful search engine are document coverage and ranking. The efficiency of lexical signatures is highly dependent on these characteristics. Since DF and PW lexical signatures extract only one or few documents in most cases, they would appear more dependent on document coverage rather than ranking. However lexical signatures such as TF which return a list of documents rely on both document coverage and ranking of search engines.

We have shown that the *unique* property of lexical signatures is not the only important factor in desired document retrieval. If we focus only on retrieving the desired documents, hybrid methods show better performance than DF and PW with *YahooGoogle*. Furthermore, DF's and PW's poor performance in extracting relevant documents when

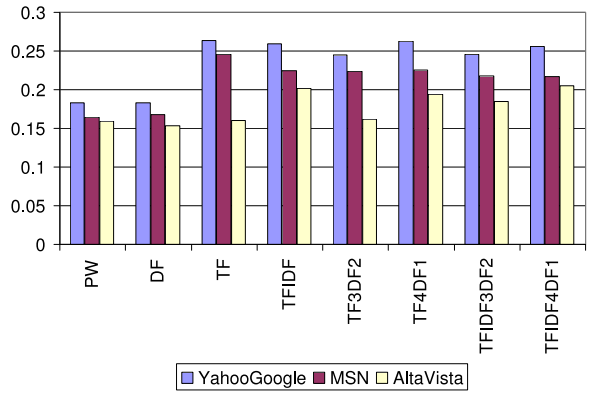


Figure 5: Average cosine value of the first ranked for *Not Retrieved Documents*

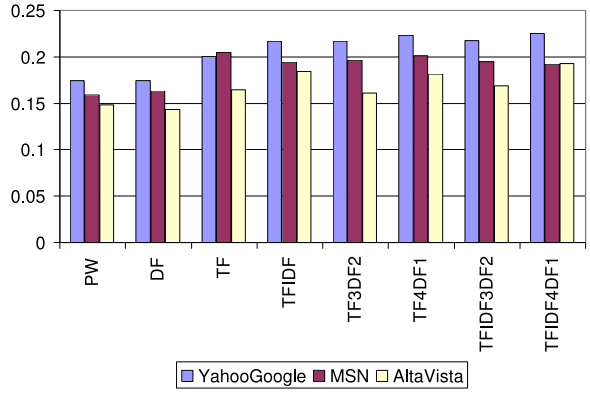


Figure 6: Average cosine value of top ten documents for *Not Retrieved Documents*

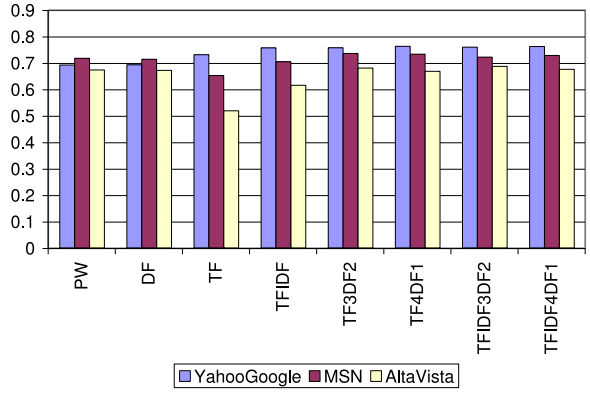


Figure 7: Average cosine value of the first ranked documents for all 980 documents

the desired one is missing make its usage for lexical signatures questionable. Unlike DF, TF is easy to compute and does not need to be updated unless the documents are modified. Also, TF extracts more relevant documents than PW and DF in the case where search engines cannot extract the desired documents. TFIDF is another good candidate for lexical signatures. Its retrieval performance on desired documents is better than TF. And when the desired document cannot be extracted, extracted documents by TFIDF are more relevant than those of DF and PW. Hybrid methods appear to be the best candidates for lexical signatures. They show excellent performance for retrieving both the desired documents and relevant documents when the desired one is missing. Also, their performance overall are more consistent than those of basic methods.

4. EMPIRICAL RESULTS FROM TREC DATA

Let's review some of the limitations of the previous experiment. First, document frequency (DF) of each word from a search engine is not the actual DF on the entire Web. Moreover, different search engines may have different document frequencies. For many reasons search engines are not easy to use for this experiment. Our data size in previous experiments is too small if we consider number of documents on the World Wide Web. Using search engines to increase our data size is prohibitive for many reasons, including search engine query policies and the changing nature of web documents.

One solution to these problems is to use TREC data for lexical signature experiments. We extracted 100,000 documents from the TREC 3, 4 and 5 data resources and did similar experiments.

4.1 Data Set and Experimental Environment

We extracted 100,000 articles from the TREC, 20,000 articles from Ziff-Davis (most articles are computer related), 40,000 articles from AP Newswire, and 20,000 articles from the Wall Street Journal and 20,000 articles from the San Jose Mercury News. We removed all tags, serial numbers that can identify the articles, dates, and names of newspapers and magazine such as *AP Newswires* in all articles. After removing stop words, we generated a term frequency list for each article and document frequency list for all articles. Our corpus contained 404,657 unique words and 219,930 words had document frequency one. Most articles were around 200 words length; Figure 8 shows the distribution of length of all articles.

We built a simple search engine for the experiment which has its own index file for all unique words, except stop words. The engine has two inputs, pageID (name of the page) and its lexical signature. When we feed a pageID and its lexical signature to the engine, it returns all pages that contain all words of the lexical signature except the target page of the given pageID. If our routine does not return any pages, the lexical signature is unique. Unlike real search engines, coverage of the search engine is complete for all documents. Since it does not have any ranking algorithm, we analyzed all returned documents by the search engine.

4.2 *Unique* Property

First, we studied the *unique* class of lexical signature methods. If a search engine does not have a ranking algorithm, this property should be the most important factor for lexical

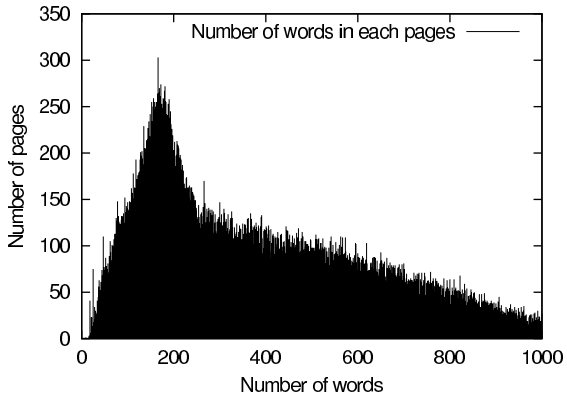


Figure 8: Distribution of number of words in each article

signatures. Figure 9 demonstrates the retrieval performance *unique* class for different lexical signature methods. As we expect, DF shows the best performance for this property and all hybrid methods have a better performance than any basic methods except DF. Since the number of documents in TREC data is much smaller than the actual web, the PW lexical signatures are more similar to the TFIDF ones. Note that hybrid methods outperform even the PW on this *unique* class.

4.3 Finding Relevant Documents

If the desired document is not returned, the best scenario is extracting the most similar document. One of properties that lexical signatures should have is robustness to minor modifications of the desired document. We assume that even though two documents are not identical, one is a modified version of the other if their similarity measure is greater than 0.9.

In Figure 10, we denote *Similar* numbers of lexical signatures that successfully found our definition of modified versions of documents. The poor performance of DF can be explained by the *uniqueness-robustness trade-off*, as Phelps and Wilensky said, i.e. if a lexical signature is chosen to minimize DF, then it would be most appropriate to extract the single desired document but its robustness for minor modification will be significantly impaired. TF3DF2 and TFIDF3DF2 show the best performance and other methods show similar results. *Fail* measures the number of lexical signatures that failed to return any documents after all words are removed. Because about half of unique words in our corpus have document frequency one, it is possible that all words in a DF lexical signature are document frequency one words. If so, no documents will be returned when the desired document is missing. For all other methods except DF, some documents are returned using lexical signatures.

Figure 11 shows the average cosine value of the best cases (most relevant returned documents) and average cases (average cosine value of all returned document) when the desired document is missing. In general, hybrid methods extract more relevant documents than basic methods and their average cosine values, except TF4DF1, are higher than those of basic methods.

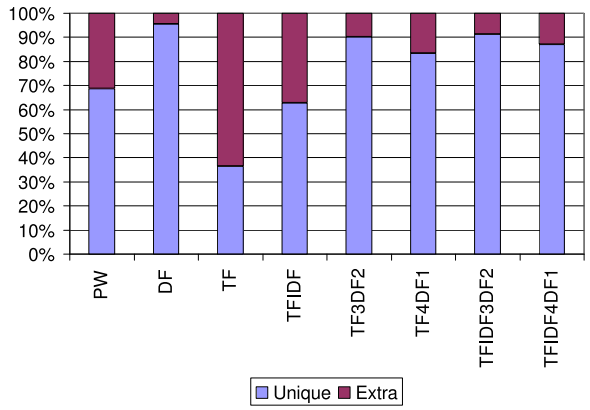


Figure 9: Retrieval performance for the unique class on TREC data.

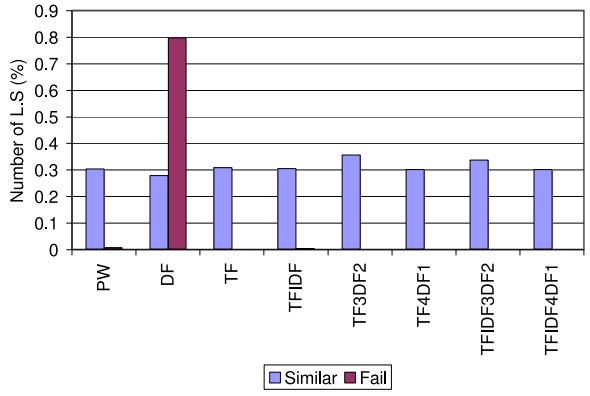


Figure 10: Number of lexical signatures that found the modified versions of documents or found no documents

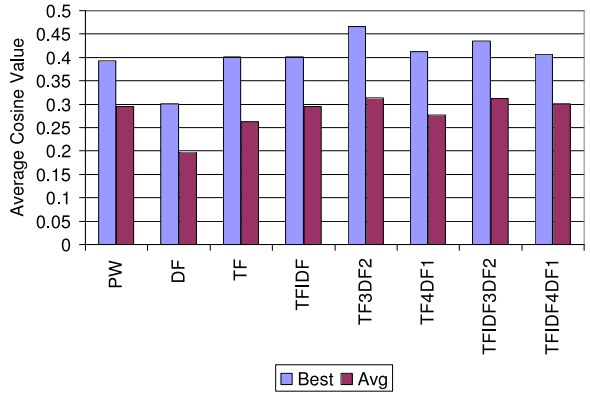


Figure 11: Average cosine value of best cases and average cases when the desired document is not extracted.

4.4 Observations

When the coverage of a search engine is complete and number of documents is relatively small (100,000 documents), hybrid methods show better performance for the *unique* extraction property than basic methods except DF. Because these methods are also excellent for retrieving relevant documents when the desired page is missing, they constitute the most promising candidates for lexical signatures. This result demonstrates that only one or two words minimizing the DF are enough for unique identification of the original document. Even though DF has the best performance for this property, it is not robust enough for minor document modifications and has poor performance for retrieving relevant documents. TF shows good performance for retrieving relevant documents when the desired document is missing but worst for the *unique* property. TFIDF and PW methods work well for both *unique* property and its *relevance*. When the number of documents in database is relatively small, PW acts similar to TFIDF.

5. CONCLUSIONS AND FUTURE WORK

5.1 Summary

Because the web does not have a well-adopted standard for maintaining persistence of information [12], the act of moving and deleting documents creates a large number of broken links throughout the web. Such broken links pose a significant problem for the growing number of people that rely on the web as a universal database. Phelps and Wilensky [14] show that even a small number of words can often uniquely identify each document on the web. In this paper, we studied eight methods for generating lexical signatures, including Phelps and Wilensky's original proposal and seven of our own. We argue that the *unique* extraction property is not the only important property for lexical signatures. As long as the desired document appears first in a returned document list, the lexical signature is effective. Also, since the coverage of search engines is limited, and documents are added, moved, modified, and deleted frequently, the ability to retrieve highly relevant documents when the desired document cannot be extracted is another important property for lexical signatures. Moreover, since different search engines have different coverages and ranking systems, the consistency of lexical signatures across search engines should be considered.

We find that DF-based lexical signatures are best at uniquely identifying documents, on both web data and TREC data. However, DF is worst at retrieving relevant documents when the desired document is missing. PW acts almost identically to DF when number of document is large, as is the case on the web. However, when the number of documents is relatively small (e.g., around 100,000 documents), PW acts like TFIDF and its relevance performance improves. TF is worst at uniquely retrieving documents, but works well for finding relevant documents. Even though TF is easy to compute and maintain, its performance variability across different search engines could outweigh its benefits. TFIDF is the best candidate for lexical signatures among the basic methods, due to its effectiveness at extracting both the desired document and relevant ones. But hybrid methods seem even better candidates for generating lexical signatures. They show good retrieval of unique documents on both web and TREC data—even better than PW on TREC data. Hybrid

methods return the desired document within the top few returned documents more often than even DF and PW. In addition, they show excellent performance in retrieving relevant documents when the desired page is missing. Finally, their ability to extract both desired and relevant documents is relatively stable over different search engines.

5.2 Performance Evaluation Methods for Search Engines

It is widely argued that search engines should be evaluated by their ability to retrieve highly relevant documents rather than all possible pages [8, 19]. Lexical signatures are good query terms that can extract relevant documents when the desired document cannot be retrieved. One limitation of lexical signatures mentioned by Phelps and Wilensky is that their performance can depend on particular search engines. However, this limitation can be exploited to evaluate search engine performance. Because a document's TF values are independent of other documents in the database, and because TF-based lexical signatures usually extract more than ten documents, the ability of TF-based signatures to extract relevant documents is highly dependent on the search engine's ranking system. By measuring similarities of returned documents with the targeted one, by using a similarity measure or human responses, we can evaluate the search engine's ability to retrieve and rank relevant documents. In our experiments, *YahooGoogle* shows the best performance (among the engines tested) for retrieving both desired documents and relevant documents, in almost all cases.

6. ACKNOWLEDGMENTS

The authors gratefully thank Dr. Gary Flake for providing valuable experimental data, and the referees and Dr. Steve Lawrence for useful comments. We gratefully acknowledge partial support from Ford Motor Co.

7. REFERENCES

[1] A. Aimar, I. Hannell, A. Khodabandeh, P. Palazzi, B. Rousseau, M. Ruggier, J. Casey, and N. Drakos. Weblinker, A Tool for Managing WWW cross-references. *Computer Networks and ISDN Systems*, 28(1&2), December 1995.

[2] K. Andrews, F. Kappe, and H. Maurer. The Hyper-G Network Information Systems. *Journal of Universal Computer Science*, 1(4):206–220, April 1995.

[3] W. Arms, C. Blanchi, and E. Overly. An Architecture for Information in Digital Libraries. *D-Lib Magazine*, February 1997.

[4] R. T. Fielding. Maintaining distributed hypertext infostructures: Welcome to MOMspider's Web. *Computer Networks and ISDN Systems*, 27(2):193–204, 1994.

[5] A. Goldberg and P. N. Yianilos. Towards an Archival Intermemory. In *Proceedings of IEEE Advances in Digital Libraries, ADL 98*, pages 147–156, Santa Barbara, CA, 1998. IEEE Computer Society.

[6] D. Ingham, S. Caughey, and M. Little. Fixing the 'Broken-Link' Problem: The W3Objects Approach. In *Computer Networks and ISDN System*, pages 1255–1268. 5th International World Wide Web Conference, May 1996.

[7] D. Ingham, M. Little, S. Caughey, and S. Shrivastava. W3Objects: Bringing Object-Oriented Technology to the Web. In *The Web Journal*, pages 89–105. 4th International World Wide Web Conference, December 1995.

[8] K. Jarvelin and J. Kekalainen. IR Evaluation Methods for Retrieving highly Relevant Documents. In *SIGIR 2000*, pages 41–48, July 2000.

[9] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, April 1998.

[10] S. Lawrence and C. L. Giles. Accessibility of information on the Web. *Nature*, 400:107–109, July 1999.

[11] S. Lawrence, C. L. Giles, and K. Bollacker. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.

[12] S. Lawrence, D. M. Pennock, G. Flake, R. Krovetz, F. M. Coetzee, E. Glover, F. A. Nielsen, A. Kruger, and C. L. Giles. Persistence of Web References in Scientific Research. *IEEE Computer*, 34(2):26–31, February 2001.

[13] G. Oberholzer and E. Wilde. Extended Link Visualization with DHTML: The Web as an open hypermedia system. Technical Report TIK-Report No. 125, Computer Engineering and Networks Laboratory (TIK), ETH Zrich, January 2002.

[14] T. A. Phelps and R. Wilensky. Robust Hyperlinks: Cheap, Everywhere, Now. In *Proceedings of Digital Documents and Electronic Publishing 2000 (DDEP00)*, September 2000.

[15] J. Pitkow. Web Characterization Activity Answers to the W3C HTTP-NGs Protocol Design Group's Questions. World Wide Web Consortium, 1998. http://www.w3.org/WCA/Reports/1998-01-PDG-answers.htm.

[16] J. E. Pitkow. Summary of WWW characterizations. *Computer Networks and ISDN Systems*, 30(1–7):551–558, 1998.

[17] K. Shafer, S. Weibel, E. Jul, and J. Fausey. Introduction to Persistent Uniform Resource Locators. In *INET 96*. Internet Society, Reston, Va., 1996.

[18] K. Sollins and L. Masinter. Functional Requirements for Uniform Resource Names. Internet Request for Comments, Dec 1994. http://ietf.org/rfc/rfc1737.txt.

[19] E. M. Voorhees. Evaluation by Highly Relevant Documents. In *SIGIR 2001*, pages 74–80, September 2001.

[20] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes 2nd Edition*. A Harcourt Science and Technology Company, 525 B Street, Suite 1990, San Diego, CA 92101 - 4495, USA, 1999.