Revisiting Lexical Signatures to (Re-)Discover Web Pages

Martin Klein and Michael L. Nelson

Old Dominion University, Department of Computer Science
Norfolk VA 23529
{mklein,mln}@cs.odu.edu

Abstract. A lexical signature (LS) is a small set of terms derived from a document that capture the "aboutness" of that document. A LS generated from a web page can be used to discover that page at a different URL as well as to find relevant pages in the Internet. From a set of randomly selected URLs we took all their copies from the Internet Archive between 1996 and 2007 and generated their LSs. We conducted an overlap analysis of terms in all LSs and found only small overlaps in the early years ($1996 - 2000$) but increasing numbers in the more recent past (from 2003 on). We measured the performance of all LSs in dependence of the number of terms they consist of. We found that LSs created more recently perform better than early LSs created between 1996 and 2000. All LSs created from year 2000 on show a similar pattern in their performance curve. Our results show that 5-, 6- and 7-term LSs perform best with returning the URLs of interest in the top ten of the result set. In about 50% of all cases these URLs are returned as the number one result and in 30% of all times we considered the URLs as not discoved.

1 Introduction

With the dynamic character of the Internet we are often confronted with the issue of missing web pages. We consider the ubiquity of "404" and "page not found" responses to be a detriment to the web browsing experience and one not adequately addressed by the Web community at large. Changes in the URL or simply discontinued domain registrations can be the reason for these negative responses but we claim that information on the web is rarely completely lost, it is just missing. In whole or in part, content is often just moving from one URL to another. As recent research has shown (2, 10, 15), we can generate lexical signatures (LSs) from potentially missing documents and feed them back into what we call the Web Infrastructure (WI) for (re-)locating these documents. The WI, explored in detail in (4, 9), includes search engines (Google, Yahoo!, MSN Live), non-profit archives (Internet Archive, European Archive) as well as large-scale academic projects (CiteSeer, NSDL). All together the WI forms the basis for this kind of "in vivo" digital preservation.

The question now arises how LSs evolve over time and how that affects their performance in (re-)discovering web pages. Figure 1 displays the scenario that

B. Christensen-Dalsgaard et al. (Eds.): ECDL 2008, LNCS 5173, pp. 371–382, 2008.

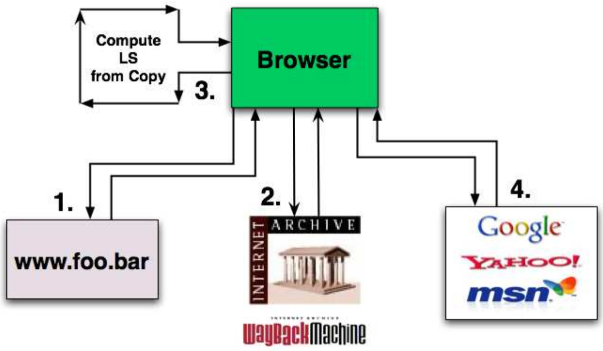


Fig. 1. Flowchart Diagram

motivated this research. In step 1 the browser requests a web page and receives a 404 error. In step 2 it queries the Internet Archive (IA) for a copy of the missing page. Since the LS is not available when the page is first noticed missing, we need to "go back in time" in the IA and query for a copy in order to generate it. Step 3 represents the process of generating a LS from the archived resource in the IA At last, in step 4, we use the LS to issue a query to one or more Internet search engines and receive the (new/correct) URL of the page that was considered missing.

With this research we conduct a baseline test where we assume web pages to be missing and use their copies from the IA to generate LSs. We submit our LSs to Google and investigate their performance by analyzing the rank of the URL of interest as a factor of the "age" of the LS. Another crucial part of this study is the composition of LSs. We distinguish between number of terms and show our experiment results with 2- to 10-term LSs. We also conducted an overlap analysis of all LSs to further investigate their evolution over time.

2 Background

A lexical signature (LS) is a small set of terms derived from a document that capture the "aboutness" of that document. It can be thought of as an extremely "lightweight" metadata description of a document as it ideally represents the most significant terms of its textual content. Table 1 shows three examples of LSs, the URLs they were generated from and the rank returned by Google (in 01/2008) along with the approximate total results. The first URL and its LS is taken from Robert Wilensky's website and is about his web page on a natural language processing project. We do not know when Wilensky generated that LS but issuing it to Google returns only that URL. If our intention is to (re-)locate the missing page only this would obviously be a very good LS. We generated the second LS in Table 1 in January 2008 and Google returned the URL as the top result along with more than 170,000 other potentially relevant results. Thus we

Table 1. Lexical Signatures generated from URLs

Rank/Total Results	**URL/Lexical Signature Terms**
1/1	http://www.cs.berkeley.edu/˜wilensky/NLP.html texttiling wilensky disambiguation subtopic iago
1/174,000	http://www.loc.gov library collections congress thomas american
na/11	http://www.dli2.nsf.gov nsdl multiagency imls testbeds extramural

still consider it a good LS since the URL is returned as the top result and the other results can be used to discover relevant pages. But with that many results it is hard to filter out the most relevant pages which makes the performance of the LS not optimal. The third LS is taken from Wilensky's and Phelps article in D-Lib Magazine from July 2000 (11). Querying Google with the LS returns 11 documents, none of which is the DLI2 homepage. The URL is indexed by Google so it should have been returned if the document was indexed with these terms but the LS is clearly dated and fails to discover the desired page.

Phelps and Wilensky (12) first proposed the use of LSs for finding content that had moved from one URL to another. Phelps and Wilensky defined a "robust hyperlink", as a URL with an LS appended as an argument such as:

```
http://www.cs.berkeley.edu/˜wilensky/NLP.html?lexical-signature=
texttiling+wilensky+disambiguation+subtopic+iago
```

where the LS is everything after the "?" in the URL. They conjectured that if the above URL would return a 404 error, the browser would look at the LS appended to the URL and submit it to a search engine to find a similar or relocated copy. Their claim was "robust hyperlinks cost just 5 words each" and their preliminary tests confirmed this. The LS length of 5 terms however was chosen somewhat arbitrarily.

Although Phelps and Wilensky's early results were promising, there were two significant limitations that prevented LSs from being widely deployed. First, they assumed web browsers would be modified to exploit LSs. Second, they required that LSs be computed a priori. It would be up to the content creator to create and maintain the LSs. Park et al. (10) expanded on their work, studying the performance of 9 different LS generation algorithms (and retaining the 5-term precedent). The performance of the algorithms depended on the intention of the search. Algorithms weighted for Term Frequency (TF; "how often does this word appear in this document?") were better at finding related pages, but the exact page would not always be in the top N results. Algorithms weighted for Inverse Document Frequency (IDF; "in how many documents does this word appear?") were better at finding the exact page but were susceptible to small changes in the document (e.g., when a misspelling is fixed). Park et al. measured the performance of LSs depending on the results returned from querying search engines and the ranking of the URL of interest in the result set. They do not compute a performance score but distinguish between four performance classes: 1) the URL of interest is the only result returned 2) the URL is not the only one returned but it is top ranked 3) the URL is not top ranked but within the top ten and 4) the URL is not returned in the top ten.

Harrison et al. (2) developed a system called Opal which uses LSs to find missing web pages using the WI. Part of their framework is the Opal server catching 404 errors and redirecting the user to the same page at its new URL or to a different page with related content.

Wan and Yang (15) explore the "WordRank" based LSs. This LS generation method takes the semantic relatedness between terms in a LS into account and chooses "the most representative and salient" terms for a LS. The authors also examined 5-term LSs only and found (similar to Park et al. (10)) that DF-based LSs are good for uniquely identifying web pages and hybrid LSs (variations of TF-IDF) perform well for retrieving the desired web pages. They claim however that WordRank- based LSs perform best for discovering highly relevant web pages in case the desired page can not be located.

Staddon et al. (13) introduce a LS-based method for web-based inference control. Following the TF-IDF method, they extract salient keywords (can be considered a LS) from private data that is intended for publication on the Internet and issue search queries for related documents. From these results they extract keywords not present in the original set of keywords which enables them to predict the likelihood of inferences. These inferences can be used to flag anonymous documents whose author may be re-identified or documents that are at risk to be (unintentionally) linked to sensitive topics.

Henzinger et al. (3) provide related web pages to TV news broadcasts using a 2-term summary (which again can be thought of as a LS). This summary is extracted from closed captions and various algorithms are used to compute the scores determining the most relevant terms. The terms are used to query a news search engine where the results must contain all of the query terms. The authors found that one-term queries return results that are too vague and three-term queries too often return zero results.

3 Experiment Design

The main objective of this experiment is to investigate the evolution of LSs over time, their term overlap and the performance of LSs in discovering their source URL. Ideally we would use snapshots of the entire web where one snapshot was taken every month over the last 15 years, generate LSs for all websites in every single snapshot and analyze their evolution. The dimensions of this scenario clearly exceeds those of our project and thus our snapshots contain only a few hundred web sites from which we derive LSs.

It is not the focus of this paper to compare the performance of LSs generated by different mathematical equations and various hybrid models (as it is done in (10)). We use the well known and understood TF-IDF based model to generate all our LSs for all web sites.

Finding a representative sample of websites is not trivial (14). For simplicity we randomly sampled 300 websites from `dmoz.org` as our initial set of URLs.

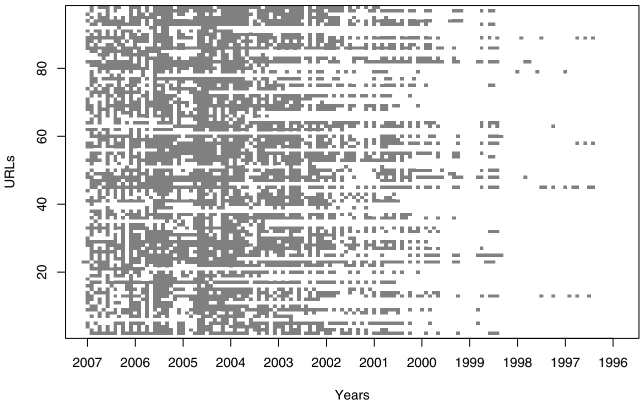


Fig. 2. Observations of all URLs from the Internet Archive from 1996 to 2007

From this pool we chose only URLs from the .com, .org, .net and .edu domain, assuming that these rather frequent domains would have a great amount of copies in the Internet Archive. The second filter we applied works similar to that of Park et al. (10). It dismisses a) all non English language websites and b) all websites with less than 50 words of textual content (HTML code excluded). This is critical because we need a good body of text to create a reasonable Lexical Signature which also is of course language specific. Our final set consists of 98 URLs (78 `.com`, 13 `.org`, 5 `.net` and 2 `.edu`).

The Internet Archive provides copies of websites from 1996 to the present. In September 2007 we downloaded all available copies of our URLs from the IA and call one copy an observation. Figure 2 shows all observations of our 98 URLs in a 12 year time span, starting in January 1996 until September 2007. The date of observation is represented on the x-axis in a monthly granularity where the mark for each year is plotted between June and July of each year. The URLs were ordered alphabetically and are numbered along the y-axis. We can see that only a few URLs actually have observations in 1996 and 1997, the earliest observation in fact was made in December of 1996 (3 URLs). The IA holds only a few observations of our sample URLs in the early years through 2000. The graph becomes more dense however from 2001 on. We also observe a 6-month period in 2005 where the number of observations decrease dramatically. We do not have an explanation for this gap but we are sharing our results with the IA in order to find the cause. Figure 2 shows an interesting fact: at any given point in time at least one of the URLs does not have an observation or, in other words, at no point in time do we have observations for *all* our sample URLs.

Generating LSs for websites following the TF-IDF scheme is not trivial. Computing IDF values requires knowledge about: 1) the size of the entire corpus (the Internet) in terms of number of documents and 2) the number of documents the term appears in. A related study (6) investigates different techniques for creating IDF values for web pages.

$$TF_{ij} \;=\; \frac{f_{ij}}{m_i} \;,\; IDF_j \;=\; \log_2\left(\frac{N}{n_j}\right) + 1 \qquad (1)$$

Equation 1 shows how we computed TF and IDF values. TF_{ij} is the term frequency of term j in document i normalized ver the maximum frequency of any term in i (m_i). IDF_j is the IDF value of term j with the total number of documents (in the corpus) N and the number of documents j occurs in n_j.

In our experiment we have copies of websites from 1996 through 2007 and want to compute their LSs. This leaves us with only one option which is to generate a "local universe" that consists of term frequencies from all downloaded websites for a particular year. Therefore we isolated the actual textual content of all websites from HTML code (including JavaScript) and created a data base of term frequencies for all terms that occur in any website of a certain year. This results in 12 term frequency data bases (1996-2007) where each of these can be considered a "local universe". For each and every single URL we aggregate all terms per year and generate LSs for each of those years. For example the URL `http://www.perfect10wines.com` has observations in the IA in 2005, 2006 and 2007 and so we generate LSs for all three years for this URL. The top ten terms of each LS along with their TF-IDF score for this URL are shown in Table 2. This example shows a core of 8 terms that occur in all three years but the ranking of the terms varies. The dynamics within the LSs, meaning the rise and fall of words can be seen with terms such as *chardonnay* (ranked 6 in 2005 and 9 in 2007) and *paso* (9 in 2005 and 3 in 2007). The example of Table 2 also shows that we did not apply stemming algorithms (*wine* and *wines*) nor eliminate stop words from the list of terms. It is left for future work to investigate the impact of stemming and stop word deletion on the LS performance.

In order to be able to compare LSs in overlap over time and their performance we generate LSs that differ in the number of terms they contain in decreasing TF-IDF order. Phelps and Wilensky as well as Park et al. chose 5-term LSs assuming 5 would be good number regarding precision and recall when feeding the LS back to Internet search engines. We chose a range from 2 terms up to 10 terms and for comparison reasons we also create 15-term LSs.

Table 2. 10-term LSs generated for `http://www.perfect10wines.com`

	2005		2006		2007	
	Term	**Score**	**Term**	**Score**	**Term**	**Score**
1	wines	8.56	wines	6.52	wines	5.25
2	perfect	5.00	wine	4.80	wine	4.50
3	wine	3.03	perfect	4.70	paso	4.50
4	10	2.60	10	3.45	perfect	4.10
5	monterey	2.24	paso	3.01	robles	3.75
6	chardonnay	2.24	robles	2.89	10	3.40
7	merlot	2.20	monterey	2.79	monterey	2.25
8	robles	1.99	chardonnay	2.79	cabernet	2.25
9	paso	1.99	ripe	1.86	chardonnay	2.25
10	blonde	1.38	vanilla	1.86	sauvignon	2.25

4 Experiment Results

4.1 Overlap Analysis of LSs

We distinguish between two different overlap measures per URL:

1. **rooted** - the overlap between the LS of the year of the first observation in the IA and all LSs of the consecutive years that URLs has been observed
2. **sliding** - the overlap between two LSs of consecutive years starting with the first year and ending with the last.

For example if an URL has copies in the IA in all years from 1996 through 2001 we would have rooted overlap values for the LSs of 1996 and 1997, 1996 and 1998, 1996 and 1999, 1996 and 2000 and finally 1996 and 2001. For the sliding overlap we have data for 1996 and 1997, 1997 and 1998, 1998 and 1999 etc. The term overlap is the number of terms two LSs have in common e.g., if two 10-term LSs have 4 terms in common its overlap would be $4/10 = 0.4$. Tables 3 and 4 show the mean overlap values of all URLs where Table 3 holds the overlap values of what was introduced as rooted overlap and Table 4 holds values for the sliding overlap. In both tables the columns represent the year of the first observation in the IA e.g., all values for all URLs with observations starting in 1996 can be found in the column headed by 1996. The mean overlap of all URLs starting in 1996 between the starting year and let's say 2001 can be thus be found in the first column and fifth row (the 2001-row) of Table 3. The overlap between 2003 and 2004 of all URLs with observations starting in 1999 can consequently be found in the fourth column (the 1999-column) and eight row (the 2003 – 2004-row) of Table 4. Due to space restrictions we only show the overlap values for 5-term LSs. We observe generally low overlap scores for the rooted overlap (Table 3). Values are usually highest in the first years after the LS was created and then drop over time. We rarely see values peaking after this initial phase which means terms once gone (not part of the LS anymore) usually do not return. This indicates that LSs decay over time and become stale within only a few years after creation. Due to the year by year comparison it is not surprising that the sliding overlap values (shown in Table 4) are higher

Table 3. Normalized Overlap of 5-Term Lexical Signatures - Rooted Overlap

compare	**Year of First Observation**										
to	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**
1997	0.33										
1998	0.13	0.33									
1999	0.13	0.20	0.56								
2000	0.13	0.33	0.49	0.51							
2001	0.20	0.27	0.31	0.46	0.58						
2002	0.13	0.33	0.33	0.32	0.48	0.64					
2003	0.13	0.13	0.40	0.40	0.47	0.54	0.66				
2004	0.13	0.13	0.36	0.35	0.40	0.53	0.60	0.66			
2005	0.13	0.07	0.38	0.37	0.37	0.42	0.50	0.63	0.58		
2006	0.13	0.20	0.31	0.35	0.38	0.48	0.51	0.46	0.62	0.80	
2007	0.20	0.20	0.27	0.29	0.37	0.44	0.50	0.37	0.52	0.60	0.90

Table 4. Normalized Overlap of 5-Term Lexical Signatures - Sliding Overlap

	Year of First Observation										
comparison	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**
1996-1997	0.33										
1997-1998	0.40	0.33									
1998-1999	0.73	0.27	0.56								
1999-2000	0.53	0.40	0.49	0.51							
2000-2001	0.47	0.87	0.56	0.62	0.58						
2001-2002	0.53	0.73	0.51	0.52	0.63	0.64					
2002-2003	0.60	0.73	0.67	0.55	0.67	0.64	0.66				
2003-2004	0.93	0.80	0.76	0.69	0.80	0.83	0.73	0.66			
2004-2005	0.87	0.80	0.73	0.66	0.82	0.68	0.83	0.74	0.58		
2005-2006	0.93	0.47	0.71	0.72	0.77	0.72	0.84	0.51	0.76	0.80	
2006-2007	0.87	0.53	0.80	0.68	0.83	0.76	0.81	0.49	0.68	0.80	0.90

than the rooted overlap values. Values often increase over time and it happens quite frequently that they peak in the more recent past. It almost seems that LSs enter a "steady state" from a certain time on. We need to point out that all values are mean values over all URLs and normalized by the maximum possible overlap. Especially for the early years due to the sparse set of observations this may be statistically unstable.

4.2 Submitting LSs to Google

We used all LSs to form queries which we issued to the Google search API between November 2007 and January 2008 and parsed the result set to identify the rank of the corresponding URLs. Search results provided by the search engine APIs do not always match the results provided by the web interfaces ((8)) but we are using the Google API for all queries and thus are not forced to handle possible inconsistencies. Since the Google API has a limit of 1000 queries per day, we only ask for the top 100 results. We distinguish between 3 cases for each URL analyzing the result set: (1) the URL is returned as the top ranked result or (2) the URLs is ranked somewhere between 1 and 100 or (3) the URL was not returned which means in our case is ranked somewhere beyond rank 100. We consider a URL for case 3 as undiscovered because as studies ((5, 7)) have shown, the vast majority of Internet users do not even click on search results beyond rank 10. We chose this classification for simplifying reasons, but are aware that there indeed is a difference between search results ranked 101..10,000 but in our study we do not distinguish between these ranks. Table 5 shows the distribution of URL ranking vs. the number of terms in the LS. It displays the relative amount of URLs returned with rank 1, ranked between 2 and 10, between 11 and 99 and beyond 100. The last row holds the mean values of all ranks. We observe a binary pattern for all n-term LSs where the great majority of all URLs return either ranked 1 or beyond 100. While the performance of 2-term LSs is rather poor, 4-term LSs seem to perform slightly better than 3-term LSs. 5-, 6- and 7-term LSs return a similar amount of URLs in the top ten with 7-term LSs returning the most top ranked results and 5-term LSs return more results ranked 2-10 and show the best mean rank. The performance of 8-, 9- and 10-term LSs is equally

Table 5. Rank vs LS Length

	Number of Terms									
Rank	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**15**
1	24.3	40.2	43.9	47.0	51.2	**54.9**	49.8	47.0	46.1	39.8
2-10	14.9	15.0	15.7	**19.4**	11.4	9.4	7.7	6.6	4.0	0.8
11-100	13.2	15.0	11.4	3.4	3.4	1.5	2.2	0.9	0.9	0.6
≥101	47.6	29.8	29.0	30.2	34.1	34.2	40.4	45.5	49.0	58.9
Mean	53.1	36.5	33.8	**32.7**	36.0	35.5	42.9	46.4	49.8	59.5

bad and worse for 15-term LSs. These results indicate that 5-, 6- and 7-term LSs all perform well. A 5-term LS seems to be the first choice when the focus is on discovering the URL somewhere in the top ten and a low mean rank. A 7-term LSs should be preferred when the focus is on finding as many URLs as possible top ranked.

LS Score Evaluation. Park et al. classified the URLs returned in four categories in order to evaluate the performance of LSs. We subsume their four categories with two continuous performance evaluation scores: **fair** and **optimistic**. Let O be the total number of observations, $R(o)$ the returned rank of one particular observation and R_{max} the maximum rank before an URL is considered undiscovered. In our experiments $R_{max} = 100$ and $R(o) \leq R_{max}$. It is important to point out that $S_{fair}(o) = 0$ and $S_{opt}(o) = 0 \; \forall \; o$ where $R(o) > R_{max}$ and $S_{fair}(o) = 1$ and $S_{opt}(o) = 1 \; \forall \; o$ where $R(o) = 1$. The equations for S_{fair} and S_{opt} are given in equations 2 and 3.

$$S_{fair}(o) = \frac{(R_{max} + 1 - R(o))}{R_{max}} \; , \; S_{fair} = \frac{\sum_{o=1}^{O} S_{fair}(o)}{O} \tag{2}$$

$$S_{opt}(o) = \frac{1}{R(o)} \; , \; S_{opt} = \frac{\sum_{o=1}^{O} S_{opt}(o)}{O} \tag{3}$$

The fair score gives credit to all URLs equally with a linear spacing between the ranks (interval measurement). For the optimistic score in contrast the distance between ranks is not equal (ordinal measurement). It comes with a huge penalty for observations in the lower ranks. For example a top ranked observation would have a score of 1 compared to another with rank 2 which would have a score of only $\frac{1}{2}$. On the other hand the optimistic score comes with a rather minor penalty between the higher ranks e.g. an observation ranked 79 with score $\frac{1}{79}$ compared to a score of $\frac{1}{80}$ for an observation ranked 80. This score optimistically expects the observations to be in the top ranks and is “disappointed” when its not, resulting in a heavy penalty. Figure 3 shows the mean values for the fair and optimistic score over all years. Here we distinguish between LSs containing $2 - 10$ and for comparison 15 terms. It also shows lines for both scores for the

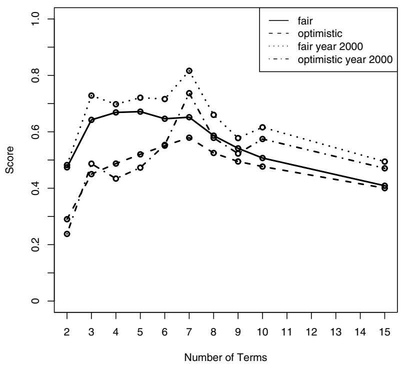
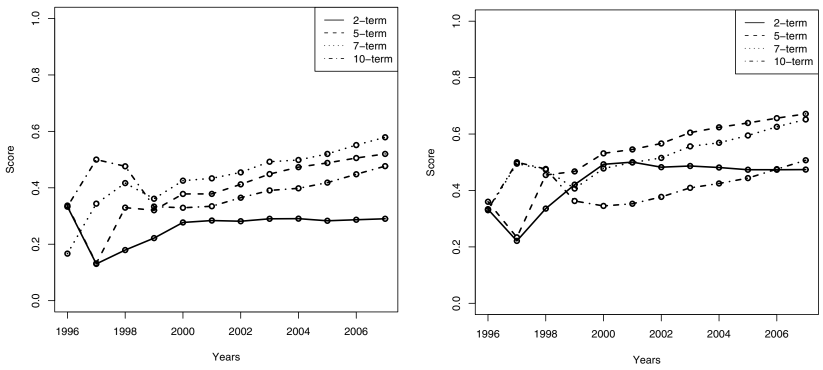


Fig. 3. LS Performance by Number of Terms



(a) S_{opt}

(b) S_{fair}

Fig. 4. Scores per year

year 2000 only as an representative example of all the years. The fair score is generally higher than the optimistic score which is surely due to the penalty between ranks implicit in the optimistic score. The high score of 5- to 7-term LSs becomes obvious and also that an increased number of terms does not gain anything, it even hurts the performance. Figure 4(b) displays the fair score of selected LSs over time and Figure 4(a) the optimistic score. Each data point represents the mean score of all URLs of a certain year (indicated by the values on the x-axis). We see the score of 5- and 7-term LSs constantly increasing reaching up to roughly 0.7 in 2007 and the low score for 2- and 10-term LSs. The optimistic score (shown in Figure 4(a)) for 2- and 10-term LSs is again low and 7-term LSs perform due to returning more top ranked URLs better than 5-term LSs. The fact that 5-term LSs have returned more URLs ranked in the top ten does not have a great impact on this type of score. The ups and downs visible in the early years are most likely due to the limited number of URLs and observations in the IA at that time. From year 2000 on we do believe to see a pattern since the lines evolve much more steadily. This may be because of an increase of observations in the IA from 2000 on (see Figure 2) in terms of more URLs observed and more copies made per day/month. Another interesting observation is the line for 2-term LSs. Regardless of its low score it shows an almost flat line from year 2000 on for both scores. A possible explanation is that 2-term LSs are in fact good for finding related pages (as shown in (3)). That means 2-term LSs constantly return relevant results with the URL of interest rarely top ranked but usually somewhere in the result set. Our intuition is that it provides good recall but poor precision explaining the low score.

5 Future Work and Conclusions

We plan to expand the scope of this exploratory research. This includes using more members of the WI for generating LSs (e.g., search more than IA in step 2 of Figure 1) and for searching for new and related versions of the document (step 4 of Figure 1).

We generated our signatures following the TF-IDF scheme and computed IDF values from "local universes" with term frequencies for each year in which our URLs were observed. In the future we can for one apply different (hybrid) models for the LS generation like introduced in (10) and for two validate the IDF values against other sources for term frequencies or grab the values from the search engine web interface like it is done in (2). A comparison study of such techniques is done in (6). We did not apply stemming algorithms nor stop word filters to the terms while generating the LSs. The impact on the LS performance could be investigated too. All these points refer to step 3 in Figure 1.

Finally a detailed analysis of the term dynamics in LSs may be conceived to be a real asset to this research. Special cases such as the treatment of a term with great significance for a certain time frame only ("one hit wonders") or dramatic changes in the overall context of a page (due to change of domain ownership, highjacked domains etc.) could be of interest when evaluating LSs over time.

The paper provides the results of our preliminary study of the performance of LSs over time. We create LSs of websites from the last 12 years, analyze their overlap with a rooted and a sliding measure, query them against an Internet search engine and evaluate the ranking of their returned URLs. Our results show that LSs decay over time. In fact the term overlap for the rooted measure decreases quickly after creation of the LS and the values for the sliding measure seem to stabilize from year 2003 on. This result indicates that LSs should not be created a priori since the content of a web page (and consequently its LS) changes dramatically over time. Now, where we have the environment to create browser extensions and plugins (like (1)) and can generate LSs from the WI as needed we can address the shortcomings in Phelps and Wilensky's work.

Regarding the number of terms we found that 2-term LSs perform rather poorly and 3- and 4-term LSs are not sufficient with slightly above 40% top ranked URLs. 5-, 6- and 7-term LSs perform best with 50% and more top ranked URLs and only about 30% undiscovered URLs. Which of these three to chose depends on the particular intention since 7 terms return the most top ranked results but 5 terms have the best mean rank. More than 7 terms have been shown to worsen the performance values. 15-term LSs e.g., show only in 40% the URLs top ranked and did not discover the URL in almost 60% of all cases.

References

1. Errorzilla - Useful error pages for Firefox, `http://roachfiend.com/archives/2006/08/28/errorzilla-useful-error-pages-for-firefox/`
2. Harrison, T.L., Nelson, M.L.: Just-in-Time Recovery of Missing Web Pages. In: Proceedings of HYPERTEXT 2006, pp. 145–156 (2006)
3. Henzinger, M., Chang, B.-W., Milch, B., Brin, S.: Query-free News Search. In: Proceedings of WWW 2003, pp. 1–10 (2003)
4. Jatowt, A., Kawai, Y., Nakamura, S., Kidawara, Y., Tanaka, K.: A Browser for Browsing the Past Web. In: Proceedings of WWW 2006, pp. 877–878 (2006)
5. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. ACM Transactions on Information Systems 25(2), 7 (2007)
6. Klein, M., Nelson, M.L.: A Comparison of Techniques for Estimating IDF Values for the Web. In: CIKM 2008 (submitted, 2008)
7. Klöckner, K., Wirschum, N., Jameson, A.: Depth- and Breadth-First Processing of Search Result Lists. In: Proceedings of CHI 2004, p. 1539 (2004)
8. McCown, F., Nelson, M.L.: Agreeing to Disagree: Search Engines and their Public Interfaces. In: Proceedings of JCDL 2007, pp. 309–318 (2007)
9. Nelson, M.L., McCown, F., Smith, J.A., Klein, M.: Using the Web Infrastructure to Preserve Web Pages. IJDL 6(4), 327–349 (2007)
10. Park, S.-T., Pennock, D.M., Giles, C.L., Krovetz, R.: Analysis of Lexical Signatures for Improving Information Persistence on the World Wide Web. ACM Transactions on Information Systems 22(4), 540–572 (2004)
11. Phelps, T.A., Wilensky, R.: Robust Hyperlinks and Locations. In: D-Lib (2000)
12. Phelps, T.A., Wilensky, R.: Robust Hyperlinks Cost Just Five Words Each. Technical report, University of California at Berkeley, Berkeley, CA, USA (2000)
13. Staddon, J., Golle, P., Zimny, B.: Web based inference detection. In: USENIX Security Symposium (2007)
14. Theall, M.: Methodologies for Crawler Based Web Surveys. Internet Research: Electronic Networking and Applications 12, 124–138 (2002)
15. Wan, X., Yang, J.: Wordrank-based Lexical Signatures for Finding Lost or Related Web Pages. In: APWeb, pp. 843–849 (2006)