A Comparison of Techniques for Estimating IDF Values to Generate Lexical Signatures for the Web

Martin Klein
Department of Computer Science
Old Dominion University
Norfolk, Virginia, USA 23529
mklein@cs.odu.edu

Michael L. Nelson
Department of Computer Science
Old Dominion University
Norfolk, Virginia, USA 23529
mln@cs.odu.edu

ABSTRACT

For bounded datasets such as the TREC Web Track the computation of term frequency (TF) and inverse document frequency (IDF) is not difficult. However, since IDF cannot be directly calculated for the entire web, it must be estimated. We see a need to estimate accurate IDF values to generate TF-IDF based lexical signatures (LSs) of web pages. Future applications for generating such LSs require a real time IDF computation. Therefore we conducted a comparison study of different methods to estimate IDF values of web pages. Our objective is to investigate how accurate these estimation methods are compared to the a baseline. We use the Google N-grams as our baseline and compare it against two IDF estimation techniques which are based on: 1) a "local universe" consisting of textual content and the according document frequencies from copies of URLs from the Internet Archive and 2) "screen scraping", a technique to query the Google web interface for document frequencies. We found a term overlap of 70 to 80% between the results of the two methods and the baseline. We further discovered a great agreement in rank correlation of TF-IDF ranked terms between our methods. Kendall τ is approximately 0.8 and the M-Score (penalizing discordances in higher ranks) is even higher, it peaks at well above 0.9. These preliminary results lead us to the conclusion that both methods are appropriate for creating accurate IDF values for web pages.

Categories and Subject Descriptors

H.3.0 [**Information Storage and Retrieval**]:

General Terms

Measurement, Performance, Design

Keywords

Inverse Document Frequency, Lexical Signature

WIDM'08, October 30, 2008, Napa Valley, California, USA.

1. INTRODUCTION

If a web page is missing (i.e., the ubiquitous HTTP 404 response) the user is often left to use search engines (SEs) to (re-)discover either the same page at its new location (e.g., the home page of a faculty member that recently transferred to another university) or a "good enough" replacement page (e.g., a different web page about the same recipe hosted by a different domain). Even if the user is capable of the (re-)discovery task, it detracts from the browsing experience.

The concept of a lexical signature (LS), a small set of terms derived from a document that capture its "aboutness", has been proposed to address this problem. An LS can be thought of as an extremely "lightweight" metadata description of a document that represents the most significant terms of its textual content. Prior work [6, 18, 23] has shown that LSs are suitable for capturing the contextual significance of web pages and can therefore, when fed back into SE interfaces, be used to successfully (re-)discover the resource. A LS is typically generated by following the Term Frequency-Inverse Document Frequency (TF-IDF) scheme. That means a list of terms ranked by their TF-IDF scores in decreasing order is generated and the LS consists of the top k terms from this list. TF ("How often does this term appear in this document?") is trivial to compute and does not depend on global knowledge of the entire corpus of documents. IDF ("In how many documents does this term appear?") in contrast also requires knowledge about the size of the entire index. In well maintained and closed datasets like the TREC Web Track [7] the computation of both values is relatively straight forward. However, this is intractable when the corpus is the entire web. Since IDF values cannot be measured, they must be estimated.

Our research plan on LSs involves a browser plugin that catches HTTP 404 responses when they occur and automatically generates a LS of the missing page. The software is supposed to provide the user with alternatives (same page at new URL and/or good enough replacement pages) which means the LS has to be generated in *real time*. The system is meant to operate on the client side which means it can not rely on a lengthy local database to look up IDF values for all possible terms since such an index would exceed the reasonable size for a plugin. With these restrictions we are motivated to find a feasible and lightweight approach to estimate IDF values for the textual content of web pages in order to generate accurate LSs.

In this paper we present the preliminary results of a study validating two different approaches to compute IDF values of

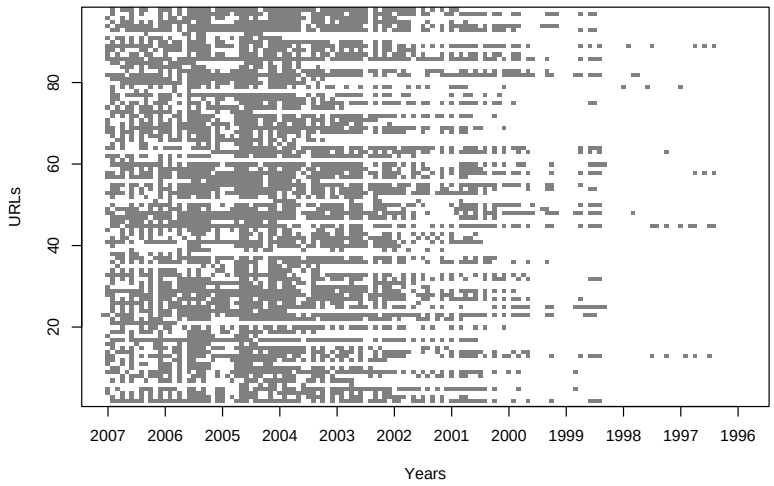

Figure 1: 10,493 **Observations of** 98 **URLs from the Internet Archive from** 1996 **to** 2007

web pages against a baseline. We use the Google N-grams [5] as our baseline for the computation of IDF values and compare it to two lightweight methods that are based on 1) a local collection of web pages and 2) "screen scraping" document frequency from SE result pages. This comparison determines whether the two alternative methods return results that are sufficiently similar to our baseline and thus warrant use for (re-)discovering web pages.

The remainder of this paper is organized as follows: in Section 2 we give a brief overview of related research and different approaches to compute IDF values. Section 3 describes the dataset used to create the local universe in detail. In Section 4 we provide an insight into our N-gram baseline IDF generation and describe the two alternative IDF computation methods which we compare to the baseline. We introduce three comparison methods to evaluate the similarity of the results of our approaches in Section 5 and provide detailed experiment results in Section 6. We conclude and address aspects for future work in Section 7.

2. RELATED WORK

2.1 TF-IDF Values from Bounded Collections

Sugiyama et al. [21] use the TREC-9 Web Track dataset [7] to estimate IDF values for web pages. The novel part of their work was to also include the content of hyperlinked neighboring pages in the TF-IDF calculation of a centroid page. They show that augmenting the generation of TF-IDF values with content of in-linked pages increases the retrieval accuracy more than augmenting TF-IDF values with content from out-linked pages. They claim that this method represents the web page's content more accurately and hence improves the retrieval performance.

Staddon et al. [20] use the British National Corpus (BNC) [13] to estimate IDF values. They introduce a LS-based method for web-based inference control. Following the TF-IDF method, they extract salient keywords from private data 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.

Ipeirotis et al. [8] introduced a method to estimate accurate DF values from databases in the web. They issue multiple queries to the database and record the number of documents returned. For queries consisting of one term only the true DF value (what the authors call *ActualDF*) is returned. For queries consisting of more than one term they record the number of documents each term occurs in and call it *SampleDF* for each term. They rank the terms by their *SampleDF* and estimate the unknown *ActualDF* values by following the distribution of rank and term frequency observed by Mandelbrot (a refinement of the Zipf distribution). They plot all *ActualDF* values and use a curve fitting algorithm to confirm their estimates. With this technique they can not only estimate accurate DF values for each term in a database but also estimate the total number of documents in the collection since the most frequent term is known to occur in a large fraction of all documents.

2.2 TF-IDF Values for Lexical Signatures

Phelps and Wilensky [19] proposed using the TF-IDF model to generate LSs of web pages and introduced "robust hyperlinks", an URL with a LS appended such as:

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

where the LS is the appended part following the "?". This example URL is taken from Robert Wilensky's website and references to a web page on one of his natural language processing projects. The LS consists of the top 5 terms in decreasing TF-IDF order. Phelps and Wilensky conjectured if the shown URL for example would return a HTTP 404 error, the web browser application could submit the appended LS

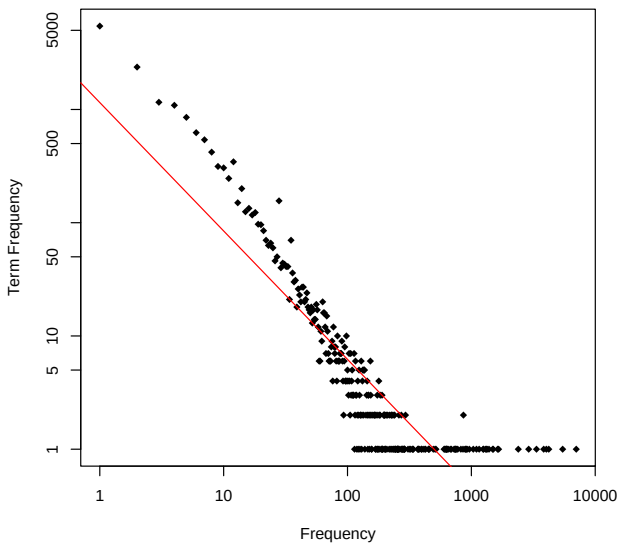

Figure 2: Term Frequency Distribution in the Local Universe

to a SE to either find a copy of the page at a different URL or a page with similar content compared to the missing page. Phelps and Wilensky did not publish details about how they determined IDF values but stated that the mandatory figures can be taken from Internet search engines.

Park et al. [18] expanded on their work, studying the performance of different variants for LS computation. They found that the performance of the algorithms depended on the intention of the search. TF-weighted algorithms were better at finding related pages, but the exact page would not always be in the top x results. Algorithms weighted for IDF on the other hand were better at finding the exact page but were susceptible to small changes in the document (e.g., when a misspelling is fixed).

Both, Phelps and Wilensky and Park et al. used search engines to determine IDF values for their LSs. They both used the "screen scraping" method with the implied assumption that the index of a search engine is representative for all Internet resources. However, they do not publish the value they used for the estimated total number of documents on the Internet. Phelps and Wilensky do not validate their screen scraping results but create LSs for several URLs and compare the retrieval performance with five different search engines. Park et al. used three different search engines to compare the retrieval performance of their LS variants. They also conducted an experiment using the TREC dataset to generate TF-IDF values and investigate the retrieval performance of their LSs in an bounded environment.

3. SELECTING THE TEST CORPUS

As shown in [22], finding a representative sample of websites is not trivial. Since it is our objective to build a local universe of web sites and our study is focused on their textual content we decided to randomly sample 300 websites from the Open Directory Project[1]. We selected only the more common domains (.com, .org, .net and .edu) from this set of URLs and dismissed all non English language URLs and all URLs which contained less than 50 terms (HTML code excluded) similar to the filter applied in [18]. Our set of URLs eventually contains 78 .com, 13 .org, 5 .net and 2 .edu URLs for a total of 98.

The Internet Archive[2] (IA) holds copies of websites beginning in 1996 to the present. The IA's crawlers traverse the web, archiving individual web pages at irregular intervals. We call each archived copy an *observation* by the IA. We downloaded all available observations of the top level page for each of our 98 websites. Our local universe consists of a total of 10493 observations, each identified by a URL (U) and the timestamp (ts) it was archived. The model of the corpus is shown in equation 1.

$$local\ universe = \left\{ \begin{array}{cccc} U_{1,ts_1} & U_{1,ts_2} & ... & U_{1,ts_n} \\ U_{2,ts_1} & U_{2,ts_2} & ... & U_{2,ts_n} \\ ... & & & \\ U_{98,ts_1} & U_{98,ts_2} & ... & U_{98,ts_n} \end{array} \right\} \quad (1)$$

Figure 1 shows all downloaded observations (in September 2007) of all 98 URLs in this 12 year span. The observation date is shown on the x-axis and the URLs, alphabetically ordered and numbered, along the y-axis. Within the 12 year time span we only see a few observations in the early years of 1996 and 1997. The graph becomes more dense however from 2000 on. The first observations were made in December 1996 and the latest in September 2007.

Figure 2 confirms that the term distribution in our local universe (despite its limited size) follows a Zipf distribution [1] of $P_k = c \times (k^{-a})$ where $a = 1.134$, similar to the distribution of English language terms. There are a total of $254,384$ terms with $16,791$ unique terms. Figure 3 shows the development of these two numbers over time where the left y-axis represents the numbers for the total terms and the right y-axis shows the numbers for the unique terms. We also computed the number of new terms per year normalized by the total number of observations in the respective year. The values range from 87.7 to 177.5 with a mean of 131.7 and a standard deviation $\sigma = 24.3$.

4. APPROACHES TO COMPUTE IDF VALUES

For the computation of IDF values two numbers are mandatory: the overall number of documents in the corpus (N) and the number of documents a particular terms appears in. We call the second value *document frequency (DF)*. Most corpora based on textual content contain all unique terms (or t-term tokens) of the corpus and the count of how many times it terms occurs in the entire corpus. We call this value *term count (TC)*.

4.1 Choice of Baseline Corpus

Table 1 gives an overview of selected corpora and their characteristics. The second column indicates what kind of documents the corpus is based upon and the rightmost column indicates whether TC values are provided with the corpus. The *TREC Web Track WT10g* is probably the most

[1] `http://www.dmoz.org/`
[2] `http://www.archive.org/`

Table 1: Available Text Corpora Characteristics

Corpus	Source	Date	Unique Terms	Number of Documents	TC
Google N-grams	Google indexed English language Web Pages	2006	$> 13M$ Terms	$> 1B$ (not available)	Available
TREC WT10g	English language Web Pages	1997	$5.8M$ [12]	$> 1.6M$	Not Available
BNC	British English Texts (newspapers, journals, books, etc.) Transcripts of Verbal Language (meetings, radio shows., etc)	1994	N/A ($100M$ Total Terms)	$4,124$	Available (from 3^{rd} party)
WaC	.uk Domain Web Pages	2006	$> 10M$	$> 2.6M$	Available

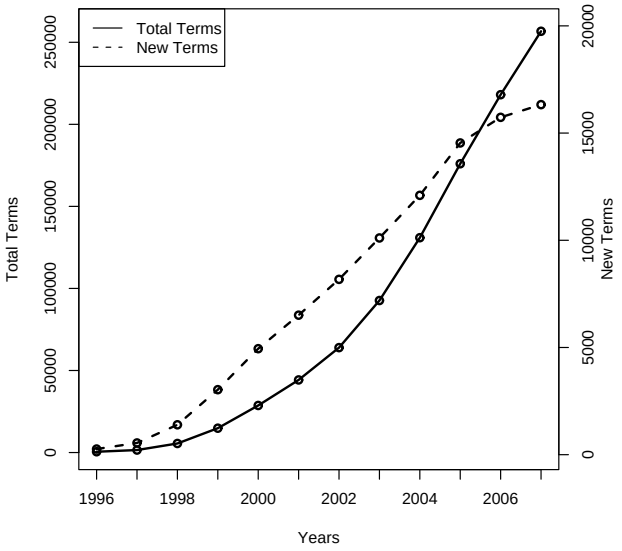

Figure 3: New vs Total Number of Terms

common and well known corpus in information retrieval research. It is based on a crawl of English language web pages but has been shown to be somewhat dated [3]. The *British National Corpus (BNC)* [13] is the only corpus in our selection that is not based on web page content but on miscellaneous written documents and transcripts of verbal communication (phone calls, interviews, etc) of British origin. The *Web as Corpus (WaC)* initiative provides its corpus for free and it exclusively consists of web pages from the *.uk* domain. The Google N-grams are attractive as a baseline for the following reasons:

1. created by a well established source - Google
2. the size of the collection seems suitable
3. the collection was created recently (2006) and
4. it is based on web pages from the Google index and thus representative of what we generate LSs from.

The N-gram dataset provides TC values for all t-term tokens ($t = 1..5$). Tokens where $t = 1$ are called unigrams and bigrams if $t = 2$. However, the dataset provides neither DF values nor original documents. Since Google is not planning to publish the DF values of the N-gram tokens we need to use a different corpus to investigate the correlation between TC and DF. We used the WaC[3] corpus in [10] for this purpose. It is freely available and its size seems suitable for our experiment. WaC provides the TC values and we are able to compute the DF values for all terms since the documents in the corpus are available. The work done in [10] shows a very strong correlation between the TC and DF values of terms from the WaC corpus. Spearman's *Rho* is ≥ 0.8 and the results are shown to be statistically significant ($p < 0.005$). Related research [17, 24, 9] also shows high correlation between TC and DF values and therefore we are led to the conclusion that the Google N-gram (TC) values can be used for accurate IDF computation. The TREC Web Track has been used to compute IDF values in [21] and the BNC in [20] but we do not know whether the authors actually computed DF values from the corpora or used TC values, which in case of the BNC are available through 3^{rd} party.

The number of documents Google used in the compilation of the N-grams is not published. Since the N-gram dataset is based on the Google index we use the last officially publicized Google index size of 8 billion (in 2005) for the second mandatory value for IDF computation[4]. The dataset only reports tokens that appear at least 200 times. For terms that do not appear in the N-gram dataset, we optimistically assume they just missed the threshold for inclusion and give them a value of 199. Since Google restricts the use of the N-grams for research purposes, we are not allowed to redistribute this data in the implementation of our software system to generate LSs. However, we can use the N-grams as our baseline to which we compare the other two approaches. Since we compute IDF values of single terms we use all 1-term tokens from the N-gram set and refer to the resulting data from the baseline as *N-gram (NG)* data.

4.2 Local Universe Data

It is clearly not feasible to download all existing web pages (language dependent), parse their content, compute and maintain the document frequency of all terms even though this would be the intuitive approach to compute accurate IDF values of web pages. However, we can download a sufficient sample set of web pages and all their copies from the last N years and use their content to create a "local universe" with document frequencies for all terms. This approach has been taken in our related research [11] which investigates the overlap and performance of LSs over time. The temporal aspect from downloading copies from the last N years

[3] `http://wacky.sslmit.unibo.it/doku.php?id=start`

[4] We have recently become aware of `http://www.worldwidewebsize.com/` and will use their estimated values in future computations.

Table 2: Top 10 TF-IDF values generated from `http://www.perfect10wines.com`

	Local Universe		Screen Scraping		N-grams	
Rank	**Term**	**TF-IDF**	**Term**	**TF-IDF**	**Term**	**TF-IDF**
1	perfect	7.77	wines	5.97	wines	7.56
2	wines	6.95	robles	5.3	perfect	7.25
3	10	6.57	perfect	4.35	robles	7.18
4	paso	6.29	paso	4.27	paso	6.93
5	wine	6.18	wine	3.26	wine	4.86
6	robles	5.4	sauvignon	3.16	10	4.52
7	sauvignon	3.54	chardonnay	3.15	chardonnay	3.99
8	cabernet	3.54	robles84	3.11	sauvignon	3.93
9	monterey	3.36	cabernet	3.09	cabernet	3.89
10	chardonnay	3.36	enthusiast85	2.91	monterey	3.49

was a focus in [11] and may not have the same significance here but it still provides a local universe with sufficient content to compare it to other sources of DF. We download all available observations of our sample set of URLs, extract all terms and store them in a local database. We do not include the content of neighboring pages. Following in- and outlinks and extract the content of neighboring pages to enhance the textual content of a target web page is a promising approach but remains for future work. This term database consequently consists of all unique terms that appear in any of the 98 web pages within the 12 year time span and the number of documents they appear in. Thus the database represents the union of the textual content of all our observations and the document frequency of all terms. With the frequencies and the total number of documents in our universe we have both mandatory values available to compute IDF values for all terms. We call the data that results from this computation *locally computed (LC)* data.

It is worth mentioning that we also generated term databases for each and every single year separately and computed IDF values for each year in respect to the according database. We found however that the results introduced in Section 6 for the per year generated IDF values were very similar to the values generated based on the LC data and thus decided not to report on them separately.

4.3 Screen Scraping the Google Web Interface

In the past, a very common approach ([18, 19, 6, 14]) to compute IDF values of web page content has been to glean document frequency information from SE result pages. This approach is known as "screen scraping" and can be done in real time which makes it attractive for many scenarios. In this approach, every single term is queried against an SE and the "Results"-value is used as the document frequency for that term. As the value for the total number of documents in the corpus we again use the last officially reported number by Google which was 8 Billion documents. Although the document frequency information is only estimated[5], this is the only information available to us. We could use the search engine APIs but the number of queries per day are limited ([16]) and the results obtained by the API may differ from the web interface results ([15]). Our screen scraping data was generated in January of 2008. In the remainder of this paper we refer to this data as *screen scraping data (SC)*.

[5]`http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=70920`

5. COMPARISON METHODS FOR TF-IDF RANKED TERM LISTS

Since we are interested in eventually generating LSs and the computation of TF and IDF is done the same way in all three approaches (including the baseline), we compare lists of terms ranked by their TF-IDF (and not only IDF) value in decreasing order.

$$TF_{ij} = \frac{f_{ij}}{m_i} \tag{2}$$

$$IDF_j = \log_2\left(\frac{N}{n_j}\right) + 1 \tag{3}$$

Equations 2 and 3 show how we computed TF and IDF values. TF_{ij} is the term frequency of term j in document i normalized over 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. We are aware of several TF-IDF variations, but we are using the method originally employed by Phelps and Wilensky [19] and also method 3 by Park et al. [18]. Phelps and Wilensky somewhat arbitrarily chose the 5-term LS and Park et al. also used 5 terms. Our earlier work [11] shows that depending on the intention (higher precision or recall) a 5- to 7-term LS is suitable. To cover a wide range of possible LSs, we transform the lists of terms ranked by their TF-IDF values into 5, 10 and 15-term LSs. Table 2 shows an example of terms in an LS and the according TF-IDF values. We took the textual content from the URL `http://www.perfect10wines.com` in 2007 and list the top 10 terms in decreasing order of their TF-IDF values for each of our three approaches. All three lists contain 10 terms but despite different TF-IDF values we only see 12 unique terms in the union of all lists. From these lists one could create a LS for the given website. The 5-term LS generated by the screen scraping method for example would be *wines, robles, perfect, paso, wine*. This example also shows that we did not apply stemming algorithms (*wines* and *wine*) nor eliminate stop words from the list of terms.

To measure the differences between these LSs we use three different comparison methods which have been proven in related research [15] to be efficient in comparing top x search engine results. The first method we apply is normalized term overlap. Since the idea is to feed the LSs back into SEs, term overlap is an important metric. For simplicity, we assume query term commutativity (although in practice the order of query terms can slightly change the results based

on proximity in the original document and other factors). We normalize the overlap of k-term lists by k. Let us for example assume list α contains three terms $\alpha = [a, b, c]$ and list β contains $\beta = [b, c, d]$. The normalized overlap would be 2/3 since α and β share two out of three terms.

In a scenario with m total terms (all candidates to make it into the LS) and an k-term LS, there is value in knowing whether a term is ranked $k + 1$, meaning it has just missed the top k, or whether it is ranked m or $m - 1$ meaning in the lower end of the ranked list of all terms. The overlap value will not reveal this information and always leave us wondering if the performance would be better if we chose a bigger or even a smaller k. For this reason we also use a modified version of Kendall τ correlation introduced by Fagin et al. [4]. This measure will answer the question since we compare LSs of different length (5, 10 and 15 terms). The common Kendall distance measure can not be applied here since its general assumption of both lists having all items in common can not be guaranteed. The modified version of Kendall does not require both lists to share all items.

Since our LSs start with the top k terms, we also think there is value in giving more credit to lists that are very similar in the high ranks and maybe less similar at the lower end than lists that show the opposite pattern. We use the M-measure introduced by Bar-Ilan [2] as our third correlation method. This method is based on an eye-tracking study showing that users are much more likely to pay attention to the top search results than to the bottom set of results. The M-Score will provide information about the locality of the term ranks in the LSs. A low score means the compared LSs show discordance in the high ranks and a high score stands for concordance in the top ranks. All three methods are normalized and return scores between 0 and 1 where 0 means complete disagreement and 1 complete agreement.

6. EXPERIMENT RESULTS

With our three models for IDF generation (*LC, SC, NG*) and the three comparison and similarity measures between ranked lists of terms in decreasing TF-IDF order we can report on all together nine comparisons. Since we consider the *NG* based data our baseline, the comparison between *LC* and *SC* is included just for completeness. It is more important for us to see how *LC* and *SC* compare against *NG*. Figure 4 displays all comparison and correlation scores between 5, 10 and 15 term LSs generated from all 98 URLs over time. The progress in time (which is due to the temporal character of the local universe) is displayed on the x-axis and the y-axis shows the appropriate values as the mean of all 98 URLs for a particular year. The first horizontal line of graphs holds the normalized term overlap scores, the second the Kendall τ scores and the third the M-Scores. The first vertical column of the plots shows the comparison values between *LC* and *NG* based data, the middle one shows *SC* and *NG* based data comparison and the graphs in the rightmost column display comparison values between *LC* and *SC* based data. The three lines visible in each of the plots represent LSs consisting of 5, 10 and 15 terms respectively.

We can observe that all nine plots look fairly similar with the lines being somewhat out of tune in the early years and from approximately year 2000 on the scores become more and more consistent with values of well above 0.5 for all three top k LSs. The noise in the early years can be explained with our sparse dataset in these years. Over time, as the dataset grows, the scores level off. We can also see the highest correlation in terms of overlap, Kendall τ and M-Score between the *SC* and *NG* based data. This is not surprising since these two datasets are supposedly based on the same (Google) index which exceeds the size of our local index (*LC* based data) by several orders of magnitude. It furthermore becomes visible that the similarity between *LC* and *NG* is greater than between *LC* and *SC*. We have two explanations for this observation. First, we argue that data returned by screen scraping is not as accurate as what Google reported in their N-grams since the Google web interface only returns estimated values for number of documents a term appears in. Second, we do have a frequency for every single term in the *NG* dataset (due to Google's threshold we assign a value of 199 to all terms that do not appear in the N-grams) but there are cases where the Google web interface does not return a value for certain terms which we consequently can not include in the LS computation.

As mentioned above we consider the term overlap (top row of Figure 4) as a very important similarity measure. After the initial noise in the early years we see an overlap of 80% in the best case (*SC* vs *NG*), about 70% in *LC* vs *NG* comparison and a worst case of 50% comparing *LC* and *SC*. The scores for Kendall τ (middle row of Figure 4) are even higher which means that not only the term overlap in the compared LSs is good but also the order of the terms is fairly similar meaning the number of pairwise disagreements within the top k LSs and thus the number of switches needed to transform one LS into the other is low.

The M-Score (displayed in the bottom row of Figure 4) accounts for the highest score in this comparison. The best cases report scores of 0.9 and above. This graph confirms that the disagreements in the LSs are rather at the low end of the ranked list of terms regardless of the method the IDF values were computed with. The M-Score is the only comparison where a slight difference between top k LSs becomes visible. Top 5 LSs seem to perform slightly worse than top 10 and top 15 especially in the comparisons *LC* vs *NG* and *LC* vs *SC*. This means the TF-IDF methods become more correlated as k increases.

All methods shown in Figure 4 comparing *LC* and *NG* as well as *SC* and *NG* data show that both, the local universe based data as well as the screen scraping based data is similar compared to our baseline, the N-gram based data. The presented scores seem to imply that the agreement between the methods for IDF computation improves or at least stays the same as we chose a greater number of terms.

7. CONCLUSIONS AND FUTURE WORK

In this paper we present the results of a study which compares two methods for the computation of IDF values of web pages with Google N-gram based values. These methods are not new to the research community and the focus of this work is to compare these methods and show their accuracy compared with a baseline. We show that there is a great similarity between local universe based data, data retrieved from parsing search engine results and N-gram data. In particular, screen scraping data from URLs from year 2000 on shows high scores for term overlap (well above 0.7) and even better scores for Kendall τ (0.8) and the M-Score with peaks above 0.9. The scores for local universe based data are still good but slightly below the screen scraping values. Term overlap is above 0.6, Kendall τ at 0.7 and the

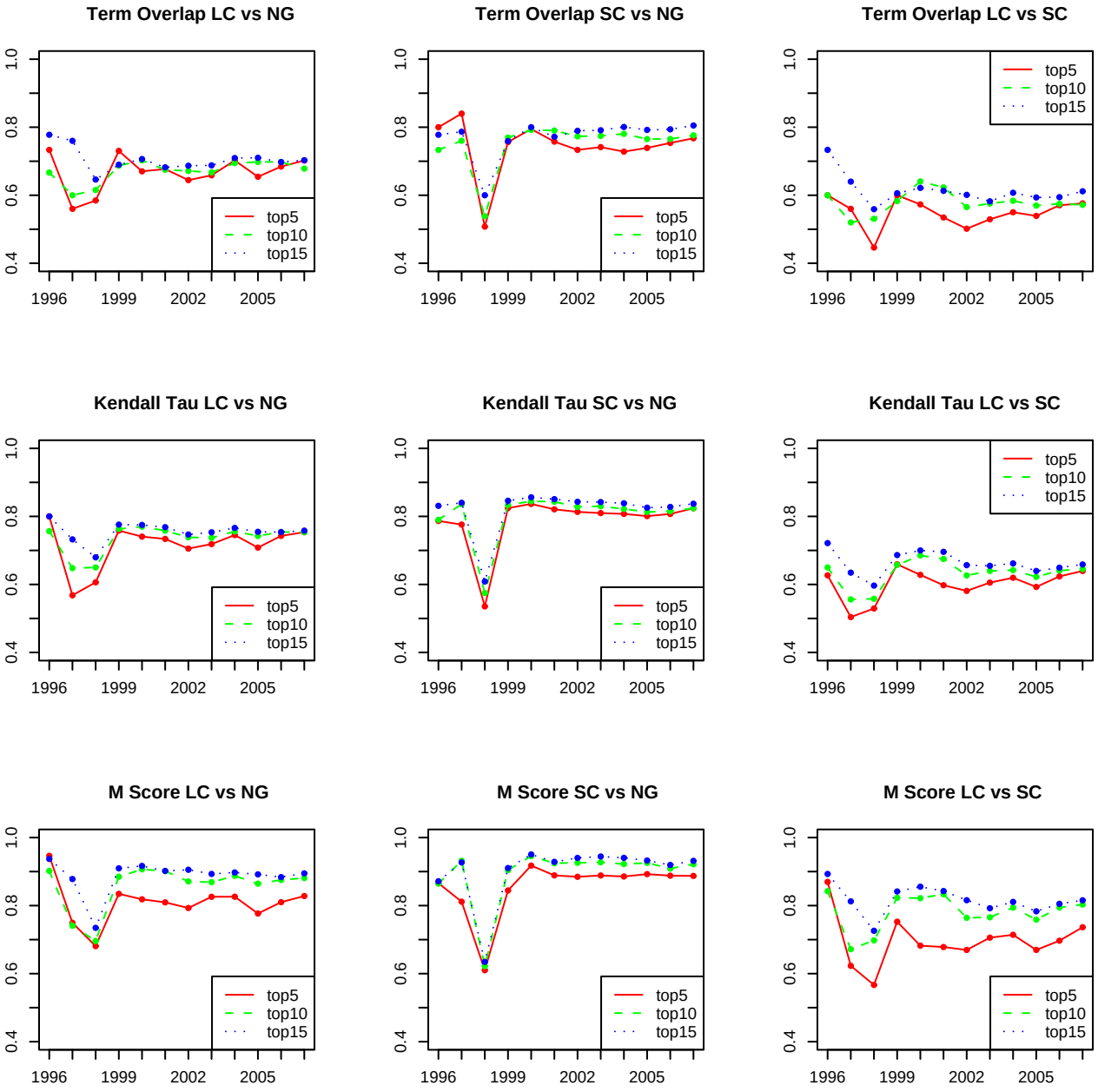

Figure 4: Term Overlap, Kendall τ and M-Score of 5, 10 and 15 Term Lexical Signatures based on the Local Universe (LC), Google Screen Scraping (SC) and the Google N-grams (NG)

M-Score has highs above 0.8. That means that both approaches provide good correlation with data derived from the Google N-grams. It also appears that with the increasing number of terms for a LS (created from the terms ranked by TF-IDF scores) the agreement between the methods also increases or at the very minimum remains stable.

We consider this data as preliminary results of our research and we see several aspects for future work. The size of our universe data is limited. A larger set of URLs and corresponding observations from the IA would be beneficial. We only used the Google web interface to retrieve the screen scraping data. It remains for future work to utilize other SEs and compare their data to the N-gram baseline. We can furthermore use other corpora such as the TREC WT10g dataset as either an alternative local universe or as an secondary baseline. Another promising experiment would be to use e.g. the TREC web collection with relevance data and compare the IR performance with our IDF estimation methods and IDF values based on the TREC corpus.

We are also designing a client-side system (i.e., browser plugin) for tracking HTTP 404 errors, generating a LS of the missing web page in real time (from the IA, SE caches, link neighborhoods) and providing the user with either a copy of the page at a different URL or alternative pages that are content-wise relevant to the missing page. Since we are not allowed to redistribute the Google N-grams data we need a different approach. This research has shown that at least two other methods are feasible for IDF generation. Screen scraping can be done in real time but needs to be done carefully since Google discourages immoderate traffic on their web interface. Local universe based data, while not difficult to generate, is probably not feasible in real time and the local universe itself (besides issues of maintenance) may exceed a reasonable size for a browser plugin. With today's ability to easily modify open source web browsers (e.g., Mozilla Firefox extensions) however we can pick up the idea of Phelps and Wilensky with LSs generated in real time and help preserve web pages.

8. ACKNOWLEDGEMENTS

We thank the Linguistic Data Consortium, University of Pennsylvania and Google, Inc. for providing the "Web 1T 5-gram Version 1" dataset. We also thank the WaCky community for providing the ukWaC dataset.

9. REFERENCES

[1] L. A. Adamic and B. A. Huberman. Zipf's Law and the Internet. *Glottometrics*, 3:143–150, 2002.

[2] J. Bar-Ilan, M. Mat-Hassan, and M. Levene. Methods for Comparing Rankings of Search Engine Results. *Computer Networks*, 50(10):1448–1463, 2006.

[3] W.-T. M. Chiang, M. Hagenbuchner, and A. C. Tsoi. The WT10G Dataset and the Evolution of the Web. In *Proceedings of WWW '05*, pages 938–939, 2005.

[4] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top k Lists. In *Proceedings of SODA '03*, pages 28–36.

[5] A. Franz and T. Brants. All Our N-Gram are Belong to You. `http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html`.

[6] T. L. Harrison and M. L. Nelson. Just-in-Time Recovery of Missing Web Pages. In *Proceedings of HYPERTEXT '06*, pages 145–156, 2006.

[7] D. Hawking. Overview of the TREC-9 Web Track. In *NIST Special Publication 500-249: TREC-9*, pages 87–102, 2001.

[8] P. G. Ipeirotis and L. Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In *Proceedings of VLDB '02*, pages 394–405. VLDB Endowment, 2002.

[9] F. Keller and M. Lapata. Using the Web to Obtain Frequencies for Unseen Bigrams. *Computational Linguistics*, 29(3):459–484, 2003.

[10] M. Klein and M. L. Nelson. Approximating Document Frequency with Term Count Values. Technical Report arXiv:0807.3755, Old Dominion University, 2008.

[11] M. Klein and M. L. Nelson. Revisiting Lexical Signatures to (Re-)Discover Web Pages. In *Proceedings of ECDL '08*, 2008.

[12] A. Kolcz, A. Chowdhury, and J. Alspector. Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization. In *Proceedings of KDD '04*, pages 605–610, 2004.

[13] G. Leech, L. P. Grayson, and A. Wilson. Word Frequencies in Written and Spoken English: based on the British National Corpus. Longman, London, 2001.

[14] Y. Ling, X. Meng, and W. Meng. Automated extraction of hit numbers from search result pages. In *Proceedings of WAIM '06*, pages 73–84, 2006.

[15] F. McCown and M. L. Nelson. Agreeing to Disagree: Search Engines and their Public Interfaces. In *Proceedings of JCDL '07*, pages 309–318, 2007.

[16] F. McCown, J. A. Smith, and M. L. Nelson. Lazy Preservation: Reconstructing Websites by Crawling the Crawlers. In *Proceedings of WIDM '06*, pages 67–74, 2006.

[17] P. Nakov and M. Hearst. A Study of Using Search Engine Page Hits as a Proxy for n-gram Frequencies. In *Proceedings of RANLP '05*, 2005.

[18] S.-T. Park, D. M. Pennock, C. L. Giles, and R. Krovetz. Analysis of Lexical Signatures for Improving Information Persistence on the World Wide Web. *ACM Transactions on Information Systems*, 22(4):540–572, 2004.

[19] T. A. Phelps and R. Wilensky. Robust Hyperlinks Cost Just Five Words Each. Technical report, University of California at Berkeley, 2000.

[20] J. Staddon, P. Golle, and B. Zimny. Web based inference detection. In *USENIX Security Symposium*, 2007.

[21] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. Refinement of TF-IDF Schemes for Web Pages using their Hyperlinked Neighboring Pages. In *Proceedings of HYPERTEXT '03*, pages 198–207.

[22] M. Theall. Methodologies for Crawler Based Web Surveys. *Internet Research: Electronic Networking and Applications*, 12:124–138, 2002.

[23] X. Wan and J. Yang. Wordrank-based Lexical Signatures for Finding Lost or Related Web Pages. In *APWeb*, pages 843–849, 2006.

[24] X. Zhu and R. Rosenfeld. Improving Trigram Language Modeling with the World Wide Web. In *Proceedings of ICASSP '01*, pages 533–536, 2001.