Exploiting Latent Semantic Relations in Highly Linked Hypertext for Information Retrieval in Wikis

Tristan Miller*
InQuira UK Ltd
50–52 Paul Street
London EC2A 4LB, United Kingdom
tmiller@inquira.com

Bertin Klein*
7P Consulting GmbH
Balcke-Dürr-Allee
40882 Ratingen, Germany
bertin.klein@7p-group.com

Elisabeth Wolf*
Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt, Germany
wolf@tk.informatik.tu-darmstadt.de

Abstract

Good hypertext writing style mandates that link texts clearly indicate the nature of the link target. While this guideline is routinely ignored in HTML, the lightweight markup languages used by wikis encourage or even force hypertext authors to use semantically appropriate link texts. This property of wiki hypertext makes it an ideal candidate for processing with latent semantic analysis, a factor analysis technique for finding latent transitive relations among natural-language documents. In this study, we design, implement, and test an LSA-based information retrieval system for wikis. Instead of a full-text index, our system indexes only link texts and document titles. Nevertheless, its precision exceeds that of a popular full-text search engine, and is comparable to that of PageRank-based systems such as Google.

Keywords

latent semantic analysis, LSA, latent semantic indexing, LSI, information retrieval, search engines, Wikipedia, wikis, hypertext, hyperlinks, large corpora

1 Introduction

1.1 Hypertext information retrieval

Traditional information retrieval systems are based on a flat, vector-space document model designed for searching plain-text corpora. Such systems are deficient when it comes to hypertext, however, as they fail to account for the topological structure of the corpus. This led to the development of the first hypertext document scoring algorithms, PageRank [22, 7, 5] and HITS [18], which score documents on the basis of the source and number of incoming and outgoing links. These scores are computed as a fixed point of a linear equation; later scoring studies [11, 6] investigated statistical or hybrid approaches.

In real-world IR applications, such as Google, the document scoring algorithm is combined with traditional vector-space IR techniques, plus metadata such as the document titles and link texts. The use of this metadata is predicated on the assumption that it is semantically related to the document under consideration. That is, the document title should reflect the document content, and likewise the text of links should accurately describe the target document—practices which are mandated by authoritative hypertext style guides [24, 10, 9, 4].

However, these guidelines are routinely ignored in HTML Web pages, for two reasons. First, few HTML authoring tools remind, encourage, or compel the writer to adhere to them. For example, in a random sample of documents from a Web corpus [15], we found that 6% of document titles were either missing, empty, a copy of the filename, or some default string such as "Untitled Document"; a further 2 to 3% were so vague as to be meaningless when read out of context. Of 2545 hyperlinks examined, 256 (10%) had no link text; 340 (13%) had link text corresponding to the URL or e-mail address of the target document; and dozens more were meaningless, referred only to navigation mechanics (e.g., "Click here"), or identified targets in a manner more suited to printed literature than to online hypertext (e.g., "Page 1").

The second reason for poor hypertext style is willful abuse of Internet search engines; authors can deliberately mislabel their documents and links in an attempt to influence page rankings for political purposes or commercial gain ("Google bombing" or "spamdexing", respectively) [2, 16]. Such manipulation is made possible by the prevailing access control structure of the Web, whereby anyone is free to post documents to be indexed, but the poster has exclusive control over their contents. This has resulted in an arms race of sorts between spamdexers and search engine developers, the former seeking new methods of artificially inflating their page ranks and the latter modifying their search algorithms to nullify those methods.

[*] The research described in this paper was carried out while the authors were at the Knowledge Management Research Group of the German Research Center for Artificial Intelligence (DFKI GmbH) in Kaiserslautern, Germany.

1.2 Wikis

In recent years, the Internet has witnessed a surge in the popularity of wikis, web applications for collaborative hypertext authoring [23]. Wikis have a number of important differences from regular HTML websites. Most significantly for the purposes of this study, wiki hypertext tends to be on-topic and have comparatively good hypertext style. This is a consequence of the following features:

Forced document validation. All wiki documents reside on a central Web server and as such must be retrieved and submitted through a Web browser. The wiki software provides a standardized editing interface, composed of HTML forms, which all users must use to submit new documents or changes to existing ones. The program which processes the form input resides on the server and can reject or fix invalid input, as well as automatically add important metadata such as the author's name and timestamp. For example, the wiki editing form will prompt the author for the document title, and if the author neglects to fill it in and submits the document anyway, the server will reject it and prompt him to correct it. By contrast, the file transfer protocol used to publish regular web pages does not enforce the semantic or syntactic validity of uploaded documents.

Lightweight markup language. Wikis accept documents not in HTML but in a lightweight markup language known as wikimarkup or wikitext [26]; the server then converts the wikitext to HTML for readers. Wikitext syntax varies from implementation to implementation, but nearly all dialects encourage or require link text to correspond to the target document title. Even where the wikitext dialect permits a disparity, the wiki software can tag the link with metadata[1] indicating the target document's true title.

Open access control structure. Wikis are designed for collaborative writing, where membership in the collaborative group can be restricted to a few named individuals or open to the general public. In the former case, abuse (in the form of spam, vandalism, or other off-topic posts) is practically nonexistent and easily rectified by blocking the perpetrator. Abuse is also low-risk in the latter case, provided the wiki is well-maintained. Popular open-access wikis such as Wikipedia are frequently policed by contributors, and any large-scale attempts at vandalism or spamdexing are immediately noticed and thwarted [25].

We posit that since wikis tend to be topically cohesive and employ stylistically correct hypertext, they do not benefit from typical Internet search engines' attempts to compensate for low-quality corpora. Any attempts to identify and suppress spamdexing, for example, can only result in false positives. Even search engines that assume their corpus documents are authoritative might not make the same assumption for the semantic correspondence between links and document titles. We therefore propose that wikis may benefit from special-purpose search engines optimized for their particular features. In particular, we believe that the semantic coherence of documents enforced by link text–document title correspondence makes wikis an excellent candidate for processing with latent semantic analysis [13, 20], a factor analysis technique which is able to discover latent semantic relations between terms and documents.

2 Algorithm

In this section we describe document indexing and search algorithms which exploit both the explicit semantic relations found in wikis and the implicit semantic relations discovered by LSA.

2.1 Document indexing

The input to the document indexing phase is a collection of wiki articles; each article is assumed to contain a unique title, plus wikitext which may contain any number of internal hyperlinks to other wiki articles. We assume that the text of a hyperlink (the *link text*) is identical to the target article's title; for wikitext dialects where this is not necessarily the case, we substitute the target article title for the link text. Each document title or link text is considered to be a single token, even if it contains multiple words.

The next and crucial step in indexing documents is to discard all text except for the title and link texts. This drastically reduces the size of the corpus—typically by 99% or more. Throwing out all this text is justified on the grounds that it is only the link texts that encapsulate the strongest semantic relations among documents, so by comparison all other information is simply noise to LSA. This is similar to the use of a stop word list to remove function words of negligible semantic content (e.g., *and*, *the*, *of*), but on a much larger scale. The net effect is essentially lossy compression of the document corpus, and allows large corpora to be indexed and searched in a tiny fraction of the storage space and time that would normally be required.

The term–document co-occurrence matrix corresponding to this reduced corpus is then tabulated, preprocessed with any desired information-theoretic transformations (e.g., tf–idf), and then dimensionality-reduced with singular value decomposition.

Figure 1 shows a sample wiki article in various forms: (a) the source code, written in a fictitious wikitext dialect where `==...==` marks the article title and `[[...]]` indicates a hyperlink; (b) how the article might be rendered in a web browser; and (c) the document vector for the article as it would appear before SVD.

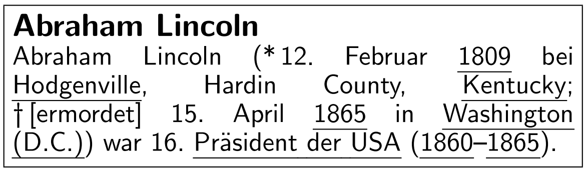
2.2 Search

A search query Q consists of one or more possibly weighted terms, represented internally as a document vector. Thus the most basic query algorithm would

[1] For example, by using the `title` attribute of the `a` element in HTML.

```
==Abraham Lincoln==
Abraham Lincoln (* 12. Februar [[1809]]
bei [[Hodgenville]], Hardin County,
[[Kentucky]]; &dagger; [ermordet] 15.
April [[1865]] in [[Washington (D.C.)]])
war 16. [[Präsident der USA]]
([[1860]]&ndash;[[1865]]).
```

(a) Wikitext source



Abraham Lincoln

Abraham Lincoln (*12. Februar 1809 bei Hodgenville, Hardin County, Kentucky; † [ermordet] 15. April 1865 in Washington (D.C.)) war 16. Präsident der USA (1860–1865).

(b) Presentation in browser

Term	Frequency
1809	1
1860	1
1865	2
Abraham␣Lincoln	1
Hodgenville	1
Kentucky	1
Präsident␣der␣USA	1
Washington␣(D.C.)	1

(c) Document vector

Fig. 1: *A sample wiki document*

be to perform pairwise vector comparisons[2] of Q with each document vector in the matrix, and return the best matches.

In the special case where Q consists of a single term which is also found in our corpus, we can use any of three additional algorithms which exploit the fact that each link text (term) in our corpus corresponds to a single document title. In the first algorithm, *Document–Document*, we find the corpus document with title Q, perform pairwise vector comparisons of its vector with each other document vector in the matrix, and return the best matches. In the second algorithm, *Link–Link*, we find the corpus term vector corresponding to Q and compare it pairwise with each other term vector in the matrix. The best-matching terms returned are link texts, but since each link text is also an article title, we return the documents with these titles. The last approach, *Link–Document*, is a hybrid of the first two, where the corpus term vector corresponding to Q is compared with each document vector.

	corpus		
	original	C_T	C_L
documents	775 696	10 419	10 419
words	195 374 109	113 019 521	658 447
links	8 196 071	N/A	658 447
kilobytes	1 891 382	775 500	6 564

Table 1: *Corpus statistics*

3 Evaluation

To test our system, we set up a user-focused experiment wherein human judges rated the relevance of search results obtained from various queries using various search engines, some of which are variations on our LSA technique and some of which are popular third-party systems.

For our document corpus, we used a subset of the German-language version of Wikipedia [1] as of 29 October 2005. The complete German Wikipedia is too large to work with efficiently for testing purposes (over 1.9 GB), so we pared it down by removing all non-articles (e.g., help and discussion pages), all leaf articles (i.e., those without any outgoing links), and all “orphan” articles of indegree < 100 (i.e., those with fewer than 100 incoming links). We refer to this corpus as C_T, as it contains the full text of the articles. From this corpus we derived a link text–only corpus C_L by removing all text outside of hyperlinks (except for the document title), plus all hyperlinks which do not target a document in the corpus. Table 1 gives some statistics on the size of our corpora.

We then selected at random three article titles which appeared in both Wikipedia’s list of featured articles of 2005 and our corpora: *Abraham Lincoln*, *Dampflokomotive* (steam locomotive), and *Todesstrafe* (death penalty). Each article title formed a search query Q which was passed to seven search engine configurations:

LSA Document–Document (LSA DD). We use LSA to compare the document with title Q to all documents in C_L, and return the top four matching document titles.

LSA Link–Link (LSA LL). We use LSA to compare the term Q to all terms (i.e, link texts) in C_L, and return the top four matching link texts.

LSA Link–Document (LSA LD). We use LSA to compare the term Q to all documents in C_L, and return the top four matching document titles.

InQuery Term–Document (IQ TD). We use the InQuery search engine [3, 8] to index C_T, submit Q as a query, and return the top four matching document titles.

InQuery Link–Document (IQ LD). We use the InQuery search engine to index C_L, submit Q as a query, and return the top four matching document titles.

Google Term–Document (Google TD). We submit Q as a query to Google with the

[2] In practice, any vector comparison function [17, 21] could be used, but in this study we use the cosine metric, which returns a similarity measure in the range $[-1, 1]$.

`site:de.wikipedia.org` directive to limit the search to German Wikipedia. Since our corpus is a subset of Wikipedia, we select the top four matching document titles which are also in our corpus.

Random outgoing links. This naïve algorithm, intended as a baseline, finds the document with title Q in C_L and returns four randomly selected outgoing hyperlink texts.

We could have obtained up to 28 unique results per topic, but since various search engines returned the same documents, we ended up with 14, 18, and 20 results for the respective topics.

We recruited 25 human judges who self-identified as fluent in German. The judges were asked to imagine that they were research assistants for three authors writing comprehensive reports on Abraham Lincoln, the steam locomotive, and the death penalty, respectively. We told them that they were to begin their research by finding relevant encyclopedia articles from Wikipedia. For each topic, we presented the judges with the combined search results for that topic. The judges were to read or skim through each Wikipedia article presented and rate them for relevance to the topic on a four-point scale from not at all relevant (1) to very relevant (4).

Implementation details

The data processing and experiments were carried out on a Sun Solaris machine using various GNU utilities, the Telcordia "Infoscale" LSA suite [12], InQuery, and a web browser for Google access.

The LSA indexing step was run with logarithmic local weighting and entropy global weighting. Since the degree of dimensionality reduction must be determined empirically, we made preliminary tests with various values and eventually settled on 1000 factors.

The three LSA-based query algorithms were implemented in a simple Bash shell script which took as input a query term, matched that term to a term or document vector from the corpus, and called the Telcordia `syn` program to perform pairwise comparisons with all the other term or document vectors. Processing each query took about three seconds of real time.

4 Results

Interjudge agreement (Pearson r) was generally very high. Mean agreements for the judges' relevance ratings for the three topics were $r = 0.747$, 0.715, and 0.767. Figure 2 is a box-and-whisker plot showing interjudge agreement for all three topics combined; the boxes delimit the first and third quartiles, the whiskers extend to the minimum and maximum, and the lines dividing the boxes show the median scores. There were no obvious outliers; we can therefore conclude that all judges basically agreed on what constituted a relevant document and that our aggregate ratings are meaningful.

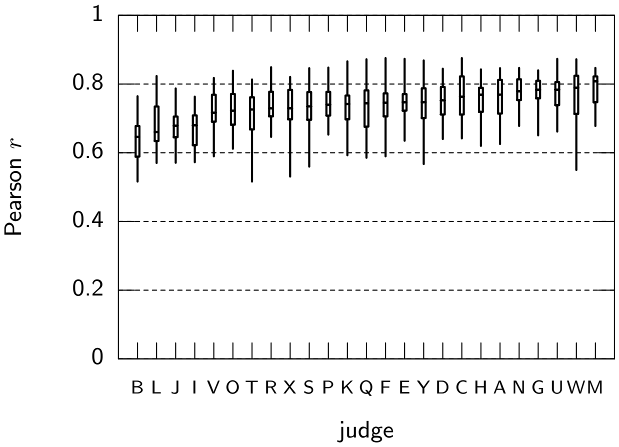


Fig. 2: *Interjudge agreement*

rank	search engine	mean relevance
1	Google TD	3.247
1	LSA LD	3.117
3	LSA LL	2.953
4	IQ TD	2.523
4	IQ LD	2.463
4	LSA DD	2.413
7	Random	1.540

Table 2: *Search engine rankings*

We next performed a two-factor repeated-measures analysis of variance (ANOVA) to determine how the choice of search engine and query affected the ratings. The two-way interaction between the search engine and query was significant at the 0.05 confidence level ($p < 0.0001$), as was the query alone ($p < 0.0001$). However, this was not entirely unexpected given the small number of queries we used in the study. Given our uniformly high interjudge agreement across queries, though, we felt justified in continuing on to perform a single-factor ANOVA across search engines. In this ANOVA, variation between groups was, of course, also statistically significant ($p < 0.0001$) so we proceeded to perform 21 pairwise t-test means comparisons. All comparisons were significant except for those between Google TD and LSA LD ($p = 0.0848$), IQ TD and IQ LD ($p = 0.3147$), IQ TD and LSA DD ($p = 0.1337$), and IQ LD and LSA DD ($p = 0.6234$). On this basis we can partition search algorithm performance into three discrete ranks, as shown in Table 2.

Figure 3 plots the combined rating scores for each of the seven search engines tested. The graph type is the same as for Figure 2, except that the dividing lines show the mean instead of the median.

As expected, the random link search algorithm performed the poorest, with a mean relevance score of 1.540. The top-ranking position is tied between Google and our LSA Link–Document algorithm. The next rank is occupied by LSA Link–Link. No statistically significant difference was observed among the InQuery Term–Document, InQuery Link–Document, and LSA Document–Document algorithms, which occupy the following rank.

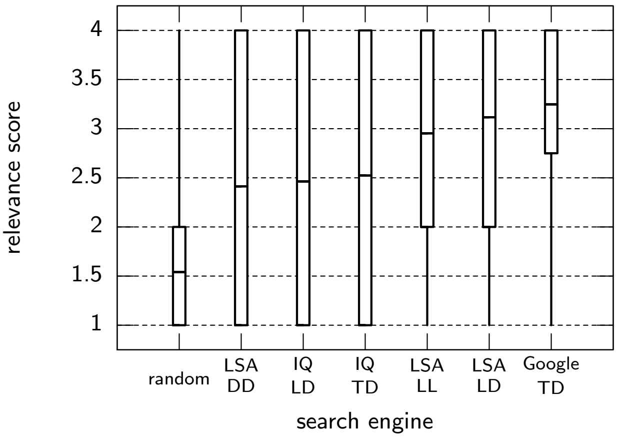


Fig. 3: *Search engine performance*

5 Conclusion

In this paper we have proposed a document indexing algorithm and family of LSA-based search algorithms which are designed to take advantage of the semantic properties of well-styled hyperlinked texts such as wikis. Performance was measured by having human judges rate the relevance of the top four search results returned by the system. When given single-term queries, our highest-performing search algorithm performs as well as the proprietary PageRank-based Google search engine, and significantly better than the non–hypertext-aware InQuery search engine. The performance with respect to Google is especially promising, given that our system operates on less than 1% of the original corpus text, whereas Google uses not only the entire corpus text but also metadata internal and external to the corpus.

Acknowledgments

The research described in this paper was supported in part by a grant (№ 01 ISC 13C) from the German Federal Ministry of Education and Research.

References

[1] *Wikipedia, die freie Enzyklopädie.* Accessed 2009-04-09 from http://de.wikipedia.org/.

[2] *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web, 14th International World Wide Web Conference*, 2005.

[3] J. Allan, J. P. Callan, W. B. Croft, L. Ballesteros, D. Byrd, R. C. Swan, and J. Xu. INQUERY does battle with TREC-6. In *Proceedings of the 6th Text REtrieval Conference (TREC-6)*, pages 169–206, 1997.

[4] T. Berners-Lee. *Style Guide for Online Hypertext*, chapter Make your (hyper)text readable. W3C, 1998.

[5] M. Bianchini, M. Gori, and F. Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 415–429, New York, NY, USA, 2001. ACM Press.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *WWW7: Proceedings of the Seventh International Conference on World Wide Web 7*, pages 107–117, Amsterdam, 1998. Elsevier Science Publishers B. V.

[8] J. Broglio, J. P. Callan, W. B. Croft, and D. W. Nachbar. Document retrieval and routing using the INQUERY system. In D. Harman, editor, *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, NIST Special Publication 500-225, pages 22–29, 1994.

[9] W. Chisholm, G. Vanderheiden, and I. Jacobs, editors. *Web Content Accessibility Guidelines 1.0*, chapter 6.13: Provide clear navigation mechanisms. W3C Recommendation. W3C, May 1999.

[10] W. Chisholm, G. Vanderheiden, and I. Jacobs, editors. *HTML Techniques for Web Content Accessibility Guidelines 1.0*, chapter 6.1: Link text. W3C Note. W3C, Nov. 2000.

[11] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 167–174, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[12] S. Deerwester. *A Brief Infoscale Tools Tutorial.* University of Chicago, second edition, 16 May 1990.

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society For Information Science*, 41:391–407, 1990.

[14] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal*, 62(6):1753–1806, 1983.

[15] Google Corp. Google programming contest, February 2002. Corpus and software. http://www.google.com/programming-contest/.

[16] S. Johnson. The art of Googlebombing. *Discover*, 25(7):22–23, July 2004.

[17] W. P. Jones and G. W. Furnas. Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38(6):420–442, 1987.

[18] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA '98: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete algorithms*, pages 668–677, Philadelphia, 1998. Society for Industrial and Applied Mathematics.

[19] A. Kontostathis and W. M. Pottenger. A mathematical view of latent semantic indexing: Tracing term co-occurrences. Technical Report LU-CSE-02-006, Computer Science and Engineering Department, Lehigh University, 2002.

[20] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2&3):259–284, 1998.

[21] M. McGill, M. Koll, and T. Noreault. An evaluation of factors affecting document ranking by information retrieval systems. Technical report, School of Information Studies, Syracuse University, Syracuse, NY, October 1979.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[23] L. Sanger et al. Wiki. In *Wikipedia, the Free Encyclopedia.* Retrieved 2009-04-09 from http://en.wikipedia.org/wiki/Wiki.

[24] A. Swartz. Don't use "click here" as link text. In W3C QA Team, editor, *Quality Tips for Webmasters.* W3C, Sept. 2001.

[25] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with *history flow* visualizations. In *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 575–582, New York, 2004. ACM Press.

[26] Wiki Markup Standard Working Group et al. WikiMarkupStandard. In *Meatball Wiki.* Retrieved 2009-04-01 from http://www.usemod.com/cgi-bin/mb.pl?WikiMarkupStandard.