International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:12, 2014

On the Interactive Search with Web Documents

Mario Kubek, Herwig Unger

***Abstract*—**Due to the large amount of information in the World Wide Web (WWW, web) and the lengthy and usually linearly ordered result lists of web search engines that do not indicate semantic relationships between their entries, the search for topically similar and related documents can become a tedious task. Especially, the process of formulating queries with proper terms representing specific information needs requires much effort from the user. This problem gets even bigger when the user's knowledge on a subject and its technical terms is not sufficient enough to do so. This article presents the new and interactive search application DocAnalyser that addresses this problem by enabling users to find similar and related web documents based on automatic query formulation and state-of-the-art search word extraction. Additionally, this tool can be used to track topics across semantically connected web documents.

***Keywords*—**DocAnalyser, interactive web search, search word extraction, query formulation, source topic detection, topic tracking.

I. Introduction

DESPITE recent advances in query formulation techniques, the most common way for users to search for information in the WWW's vast amount of data is to express their information needs by entering search words into the query input fields of well-known web search engines such as Yahoo! or Google. The search words are chosen such that they are likely to appear in documents that fulfil those information needs. Here, the challenge is to describe desired contents with only a few expected keywords and submit them as queries in order to find matching web documents within the search engines' huge document indexes. Usually, this method results in a large number of documents presented to the users, who mostly review only the first 10 to 30 results. Algorithms like PageRank [1] try to ensure that the most relevant search results appear among those first 30 result entries. Thus, the search for non-trivial information in the web becomes an iterative process. In a first step, the user formulates a query and sends its search words to a search engine, which mostly serve as contextual information. A few returned results are inspected afterwards in order to specify the wanted information more precisely and to find out more specific keywords from the target topic. Then, the newly found terms are used for another request and the described process continues until the desired documents are found. Depending on the experience of the user, this process may take more or less time and may become especially hard and tedious when

- the amount of returned search results is too large,
- the search results cover too many topics or subtopics,
- the user lacks sufficient knowledge on the topic of interest and
- completely new or emerging topics are of interest for which there might not yet exist a proper terminology.

An automatic recommendation of search words and queries that regards the context of the search subject would therefore be of great help to the user. Consequently, in several research projects, methods have been developed to identify and present additional search words and search word alternatives to the user during this process. Google itself uses a statistical approach [2] and offers frequently co-occurring terms from its users' queries as suggestions to refine queries. This method of course fails if completely new or very seldom requested content is searched. However, obtained information about the user is often used for other purposes than just for the improvement of web search, e.g. to place targeted commercial advertisement based on the area of his or her search activities.

Therefore, other (and mostly scientific) approaches apply locally working methods. They analyse local documents in order to provide a more precise description of the user's information needs in form of appropriate keywords for the current search context and do not transfer any user related information to the (centralised) search engine. One example is the application FXResearcher [3]. This tool carries out a text analysis of a set of documents kept in specific user-defined directories in order to extract additional keywords for the next search iterations. In addition, recently downloaded and evaluated documents may be used for result improvements in the next steps.

Additionally, most search engines today allow a search for pictures with amazing results. Either the content of pictures must be described by so called metadata (which are again keywords describing contents and contexts) [4] or a search for similar pictures (using their colour, contrast and other hard image values) [5] is provided. The tool PDSearch [6] uses a different approach. It has been designed and developed to support a search for web documents using the textual context (e.g. URL and description) of pictures in web pages that the user selects. This textual context is automatically extracted, analysed and used to generate proper queries to search for similar web documents. Moreover, PDSearch can carry out this task for a number of selected pictures in one search step.

In this article, the interactive search application DocAnalyser is presented which provides solutions to the listed problems. It enables users to find similar and related web documents based on automatic query formulation and state-of-the-art search word extraction. DocAnalyser uses the locally displayed contents of currently visited webpages for this purpose.

The remaining paper is structured as follows: in the next

Dr.-Ing. Mario Kubek and Prof. Dr.-Ing. habil. Herwig Unger are with the Department of Communication Networks at the FernUniversität in Hagen, 58084 Hagen, Germany (phone: +49 23319871155; e-mail: kn.wissenschaftler@fernuni-hagen.de).

section, DocAnalyser's HITS-based algorithm for search word extraction is outlined. Section three focuses on the use cases of this application. Afterwards, section four discusses its planned utilisation as a fingerpost in the web to guide users to relevant contents matching their interests and information needs. Section five concludes the paper and discusses future application fields of DocAnalyser.

II. Extracting Keywords and Source Topics Using Extended HITS

In recent years, state-of-the-art and graph-based algorithms for keyword extraction have been proposed [7]-[9] that outperform classic and well-known methods such as TF-IDF [10] and difference analysis [11]. That is why, in DocAnalyser, a new graph-based method for keyword extraction presented in [12] has been implemented. The next subsections outline the idea behind this method which relies on the analysis of directed co-occurrence graphs consisting of a text's terms and their relations.

In order to be able to find not only similar web documents using their keywords (main topics) as search words, but to also identify related web documents that deal with non-obvious, yet influential aspects of the analysed texts, the notion of source topics is introduced. These terms strongly influence the main topics in texts and represent their inherent concepts, yet are not necessarily important keywords themselves. They are especially helpful when it comes to applications like tracking topics.

The HITS algorithm [13], which was initially designed to evaluate the relative importance of nodes in web graphs (which are directed), returns two lists of nodes: authorities and hubs. Authorities are nodes that are often linked to by many other nodes. They can be determined using the [1] algorithm, too. Hubs are nodes that link to many other nodes and therefore topically influence the nodes they link to. Nodes are assigned both an authority and hub value to evaluate their centrality. For undirected graphs, the authority and the hub scores of a node will be the same, which is naturally not the case for the web graph.

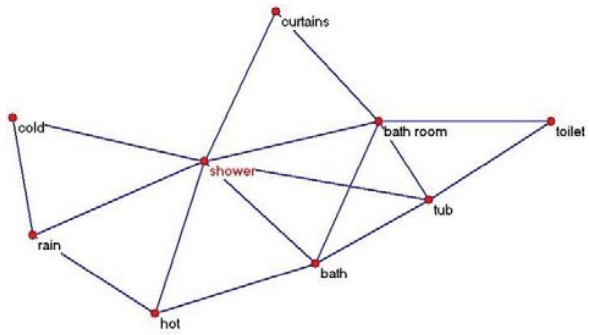

Fig. 1 A Co-occurrence graph for the word "shower"

As shown in [12], the HITS algorithm can also be used to analyse word nets such as co-occurrence graphs. Here, the gained authorities can be regarded as the characteristic terms or keywords of the analysed text, whereas the hubs represent its source topics. However, co-occurrence graphs are usually undirected which is suitable for the flat visualisation of term relations (Fig. 1) and for applications like query expansion via spreading activation techniques [3].

For the mentioned requirement to extract keywords as well as source topics, these graphs must be directed such that individual term associations of different strengths can be taken into account and lead to different authority and hub scores for each term. The idea behind this approach is that a random walker is likely to follow links in co-occurrence graphs that lead to terms that can be easily associated with the current term he is visiting. Nodes that represent terms that are linked with a low association value, however, should not be visited very often. This also means that nodes that reside on paths with terms linked with high association values should be ranked highly as they are more likely to be visited.

As explained in [12], in order to determine the needed term associations, (1) can be applied:

$$Assn\ A \rightarrow B\ = \frac{A \cap B}{n_{max}} \tag{1}$$

Here, $A \cap B$ is the number of times terms A and B co-occurred in the text on sentence level and n_{max} is the number of sentences any term occurred in. The resulting weight is derived by multiplying the term $\frac{A \cap B}{A}$ that indicates the basic association strength of A with B by the term $\frac{A}{n_{max}}$ that accounts for the relative frequency of A (on sentence level) in the text as a measure for A's overall importance. Here, A is the number of sentences term A occurred in.

Also, it is sensible to only take into account the direction of the dominant association (the one with the higher basic association strength) to generate a directed co-occurrence graph to properly support the following analysis. A relation of term A with term B obtained this way can be interpreted as a recommendation of A for B when the association strength is high. Relations gained by this means are more specific than undirected relations between terms because of their direction. They resemble a hyperlink on a website to another one. In this case, however, this link has not been manually and explicitly set and carries an additional weight that indicates the strength of the term association. The set of all such relations obtained from a text represents a directed co-occurrence graph ready to be analysed by the extended HITS algorithm. In order to take the term association values *Assn* into account, the formulas for the update rules of the HITS algorithm must be extended. The authority value of a node x can then be determined using (2):

$$a\ x\ =\ \ _{v \rightarrow x} h\ v\ \cdot Assn(v \rightarrow x) \tag{2}$$

The hub value of a node x can be calculated using (3):

$$h\ x\ =\ \ _{x \rightarrow w} a\ w\ \cdot Assn\ x \rightarrow w \tag{3}$$

The following steps are necessary to obtain two lists containing the analysed text's authorities and hubs based on

these update rules:

1. Remove stopwords and apply stemming algorithm on all terms in the text. (Optional)
2. Determine its directed co-occurrence graph G based on one of the solutions presented in the previous subsection.
3. Determine the authority value a(x) and the hub value h(x) iteratively for all nodes x in G using (2) and (3) until convergence is reached (the calculated values do not change significantly in two consecutive iterations) or a fixed number of iterations has been executed.
4. Return all nodes in descending order by their authority and hub values with their representing terms and their authority and hub values.

These term lists (can be regarded as term clusters, too) can also show, however, an overlap when analysing directed co-occurrence graphs. Hence, this is a soft graph clustering technique. In each case, these lists are ordered according to their terms' centrality scores. As an example, Table I presents for the article "Love" from the English Wikipedia these two extracted lists, whereby the following parameters have been applied:

- removal of stopwords
- restriction to nouns and adjectives
- base form reduction
- activated phrase detection

TABLE I
TERMS AND PHRASES WITH HIGH AUTHORITY AND HUB VALUES OF THE WIKIPEDIA-ARTICLE "LOVE"

Term	*Authority value*	*Term/Phrase*	*Hub value*
love	0.54	friendship	0.21
human	0.30	intimacy	0.18
god	0.29	passion	0.16
attachment	0.26	religion	0.14
word	0.21	attraction	0.14
form	0.21	platonic love	0.13
life	0.20	interpersonal love	0.13
feel	0.18	heart	0.13
people	0.17	family	0.13
buddhism	0.14	relationship	0.12

The example shows that the extended HITS algorithm can determine clusters of the most characteristic terms (authorities) and source topics (hubs). Especially the list of source topics provides valuable insight into the analysed text's topical background which is useful to find related (not necessarily similar) content when used in queries. Another empirical finding was that the quality of the authority and hub lists improved when analysing clusters of semantically similar documents instead of single texts. The reason for this is that by using a larger textual basis for the analysis the calculated term association values are more statistically meaningful.

III. DOCANALYSER

This section introduces the new interactive web-based search application DocAnalyser [14]. Its usage and implementational details will be discussed in the next subsection. Afterwards, the use cases to search for similar and related web documents will be elaborated on in detail.

A. Usage and Implementation

DocAnalyser [14] is a new web service that offers a novel way to interactively search for similar and related web documents and to track topics without the need to enter queries manually. The user just needs to provide a web content using the DocAnalyser's bookmarklet (downloadable JavaScript code in a web browser bookmark to send the selected web content to the DocAnalyser web service) to be analysed. This is usually the web page (or a selected part of it) currently viewed in the web browser (see Fig. 2).

Fig. 2 Selection of Web Content to be Analysed

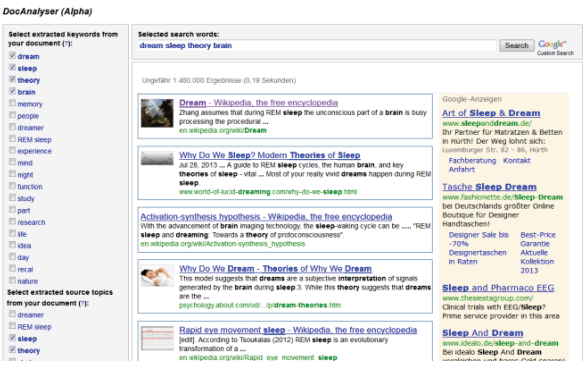

Fig. 3 DocAnalyser's Result Page

DocAnalyser then extracts its main topics and their sources (important inherent, influential aspects / basics) as presented in the previous section and automatically uses (at most four) of them as search words. The returned search words and web search results are generally of high quality. Usually, the currently analysed web document or content is found again among the Top-10 search results which underlines and confirms this statement. Therefore, this tool can also be used to some extent to detect plagiarism in the WWW. The user can also easily modify the preselected query containing the most important keywords by clicking on them on the term lists or by using the query input field. Fig. 3 shows a screenshot of DocAnalyser with extracted search words (keywords and source topics) from the Wikipedia-article "Dream" and Google's returned web results.

In its current Java-based implementation, DocAnalyser offers the following features:

- handling of different file formats such as PDF, DOC, DOCX, PPT, PPTX, HTML, XML and TXT
- language detection (German/English) and sentence

splitting
- part-of-speech (POS) tagging (nouns, verbs, adjectives, adverbs, card.)
- phrase detection based on POS filters
- base form reduction
- high-quality keyword and source topic extraction based on the analysis of the selected content's directed co-occurrence graphs

Another advantage of DocAnalyser's multi-threaded server is its ability to handle multiple user requests at a time. The graphical user interface of the result page is compatible with any modern web browser.

B. Discussion

The service DocAnalyser offers goes beyond a simple search for similar documents as it presents a new way to search for related documents and documents with background information using source topics, too. This functionality can be regarded as a useful addition to services such as Google Scholar (http://scholar.google.com/), which offers users the possibility to search for similar scientific articles. It becomes especially useful when an in-depth research on a topic of interest has to be conducted. Since the implemented extended HITS-based algorithm for keyword extraction employs semantic term relations for doing so, the quality of the obtained search words and the web search results gained using them is also high. Another noteworthy advantage of this keyword extraction algorithm is that it works on single texts as it does not rely on solutions based on term frequencies such as TF-IDF [10] and difference analysis [11] that require preferably large reference corpora.

As indicated before, in order to provide instant results, the (at most) four most important keywords/key phrases are automatically selected as search words which are sent to Google. The author chose four terms because, as discussed in [15], 4-term co-occurrences (when used as search words) provide the best trade-off between a precise search context description and a useful amount of search results. In this case, however, this 4-term co-occurrence (the query) is constructed for the complete web content provided, not only for a text fragment such as a sentence or paragraph. Although this approach yields useful results, the user still has the option to interactively modify the preselected query. The extracted source topics provide another interesting use case, viz, the possibility to track topics to their roots by iteratively using them as search words. This use case along with further application scenarios of DocAnalyser will be explained in the next section.

IV. Further Application Scenarios

In this section, further interesting application fields to make use of the presented keyword extraction algorithm and possible ways to extend DocAnalyser's use cases will be explained. Here, the focus will be put on approaches to track topics using an analysed document's source topics.

A. Topic Tracking Using Source Topics

As the source topics of a text document represent its set of important influential aspects, the idea seems natural to (iteratively) use them as search words to find more documents in the WWW that mainly deal with these topics when an in-depth research on a topic of interest needs to be conducted. This way, it is possible to find related, not necessarily similar documents, too. The reason for this is the naturally occurring topic drift in the results induced by the semantic differences between authorities and hubs which become obvious when they are used as search words. Therefore, source topics of documents can be used as means to follow topics across several related documents. DocAnalyser implicitly supports this kind of search. A possibility to improve the quality of the returned web search results in a future implementation would be to topically cluster the source topics prior to using them as search words. Here, standard cluster algorithms such as K-means [16] could be helpful. The needed input parameter k to specify the number of term clusters to be generated could be set according to the number of source topics that are most dissimilar to the other source topics (the similarity of two terms can be determined by counting and weighting the overlap between their set of co-occurring terms, e.g. using measures such as cosine similarity). Also, graph-based cluster algorithms such as the so-called Chinese-Whispers-algorithm [17] could be used for this purpose as it seems natural to employ such kinds of algorithms on the already calculated directed co-occurrence graph. This way, queries containing terms from specific topical clusters could be automatically offered to users while indicating their different topical orientations. This would present a major step forward to realise a system acting as an automatic fingerpost guiding users to relevant information in the web.

B. Automatic Link Induction

Another interesting application for the utilisation of detected source topics and keywords can be seen in the automatic linking of related documents found in large corpora like the WWW. If a document A primarily deals with the source topics of another document B, then a link from A to B can be set. This way, the herein described approach to obtain directed term associations is modified to gain the same effect on document level, namely to calculate recommendations for specific documents. A search agent or web search engine could build up an internal index of these document relations. By doing so, it incrementally learns new document relationships. New search results can then be provided with links to similar but also to related documents that primarily deal with their source topics in order to give users access to background information on a topic of interest and to also follow topics across multiple documents.

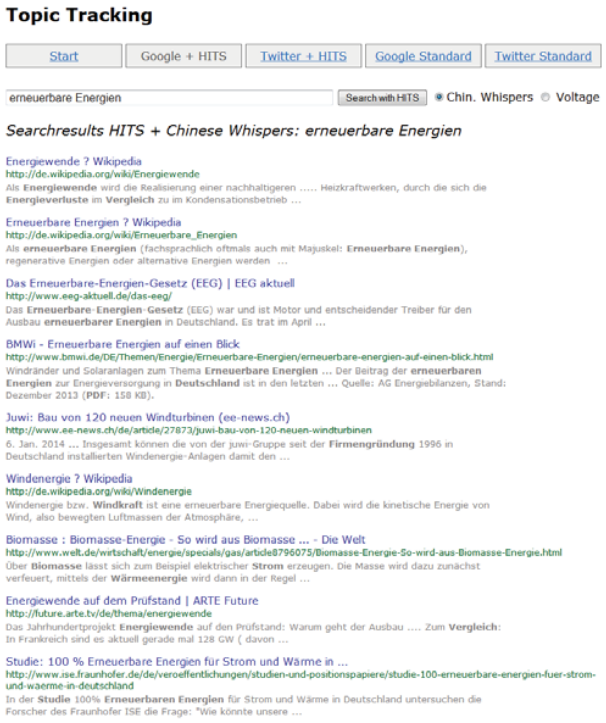

Fig. 4 Tracking the Topic: "Renewable Energies" (German: erneuerbare Energien)

The idea behind this approach is that especially the determined source topics can lead users to documents that cover important aspects of their analysed and presented search results. A first realisation of this idea in combination with the clustering approach outlined in the previous subsection has been presented and evaluated in [18]. In Fig. 4, a screenshot of the graphical user interface of this implementation along with an example result list for the query "renewable energies" (German: erneuerbare Energien) is given. The result page shows more topically specific entries (covering biomass and wind power) than Google's initial result list for this query. Further conducted experiments also confirm the validity of this new approach for topic tracking.

C. Re-Ranking of Web Search Results

These automatically generated links between web search results can also be very useful in terms of positively influencing the ranking of search results, because these links represent semantic relations between documents that have been verified in contrast to manually set links e.g. on websites, which in turn, additionally, can be automatically evaluated regarding their validity using this approach for automatic link induction, too. Based on their found relationships, web search results could be reordered in such a manner, that topical clusters become visible. Also, by comparing the newly found documents' term lists with the lists of keywords and source topics of locally existing documents e.g. using local search agents such as FXResearcher [3], it is possible to re-rank them based on their similarity with the local knowledge. As this function will take a possibly large amount of time, its use is not appropriate when a timely response is needed. However, it is sensible, when an in-depth analysis of a topic is required and real-time demands play a secondary role.

V. Conclusion

The new web-based application DocAnalyser to interactively search for similar and related web documents has been introduced. Its features and operation have been explained in detail. The main benefit DocAnalyser provides is the automatic formulation of Google queries containing keywords and source topics of analysed documents, the only required input parameter. Emphasis has been put on the description of the applied HITS-based algorithm for automatic search word extraction and the generation of directed co-occurrence graphs needed for its utilisation. The derived use case of tracking topics in the WWW using these techniques has been extensively discussed, too. Even so, DocAnalyser's solutions and approaches are just a first step towards a new way of searching the WWW, viz, the technically supported guidance of users to web documents and contents that actually fulfil their information needs. The author's future research will focus on the appropriate extension of these solutions.

References

[1] L. Page, S. Brin, R. Motwani, T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", Technical Report, Stanford Digital Library Technologies Project, 1998.
[2] Website of Google Autocomplete, Web Search Help, https://support.google.com/websearch/answer/106230
[3] M. Kubek, H.F. Witschel, "Searching the Web by Using the Knowledge in Local Text Documents", In Proceedings of Mallorca Workshop 2010 Autonomous Systems, Shaker Verlag, Aachen, 2010.
[4] K. Yee, K. Swearingen, K. Li, M. Hearst, "Faceted Metadata for Image Search and Browsing", *CHI '03 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 401–408, New York, 2003.
[5] F. Tushabe, M. H. Wilkinson, "Content-based Image Retrieval Using Combined 2D Attribute Pattern Spectra", *Advances in Multilingual and Multimodal Information Retrieval*, pp. 554–561, Springer, Heidelberg, 2008.
[6] P. Sukjit, M. Kubek, T. Böhme, H. Unger, "PDSearch: Using Pictures as Queries", *Recent Advances in Information and Communication Technology*, Advances in Intelligent Systems and Computing, Vol. 265, pp. 255–262, Springer International Publishing, 2014.
[7] J. Wang, J. Liu, C. Wang, "Keyword Extraction Based on PageRank", *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, Vol. 4426, pp. 857–864, Springer Berlin Heidelberg, 2007.
[8] R. Mihalcea, P. Tarau, "TextRank: Bringing Order into Texts", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pp. 401–411, Association for Computational Linguistics, 2004.
[9] M. Kubek, H. Unger, "Search Word Extraction Using Extended PageRank Calculations", *Autonomous Systems: Developments and Trends*, Volume 391 of Studies in Computational Intelligence, pp. 325–337, Springer Berlin Heidelberg, 2011.
[10] G. Salton, A. Wong, C.S. Yang, "A vector space model for automatic indexing", *Communications. of the ACM*, Vol. 18, Issue 11, pp. 613–620, 1975.
[11] G. Heyer, U. Quasthoff, T. Wittig, *Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse*, W3L-Verlag, 2006.
[12] M. Kubek, "Dezentrale, kontextbasierte Steuerung der Suche im Internet", PhD Thesis, FernUniversität in Hagen, 2012.
[13] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment", *Proc. of ACM-SIAM Symp.on Discrete Algorithms*, San Francisco, California, pp. 668–677, 1998.

[14] Website of DocAnalyser, http://www.docanalyser.de, 2014, Last retrieved on 10/01/2014
[15] M. Kubek, H. Unger, "On N-term Co-occurrences", *Recent Advances in Information and Communication Technology*, Advances in Intelligent Systems and Computing, Vol. 265, pp. 63–72, Springer International Publishing, 2014.
[16] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281–297, University of California Press, 1967.
[17] C. Biemann, "Chinese Whispers: An Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems", *Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-06*, pp. 73–80, ACL, New York City, 2006.
[18] V. Heß, "Implementierung und Evaluation eines Verfahrens zur Themenverfolgung in großen Korpora", Master's thesis, FernUniversit¨at in Hagen, 2014.