Mario Kubek
Technical University of Ilmenau, Germany
mario.kubek@tu-ilmenau.de
Hans Friedrich Witschel
University of Leipzig, Germany
witschel@informatik.uni-leipzig.de

A SYSTEM FOR AUTOMATIC QUERY EXPANSION IN A BROWSER-BASED ENVIRONMENT

Abstract: This paper introduces a system for automatic query expansion. It is aimed at helping users to search for text documents in the World Wide Web through the usage of local text corpora residing on the user's computer. These text documents are used to select appropriate terms to expand search queries. With this technology the search context is not just generated on the server side of a web search engine, but emerges right on the user's PC. This helps to consider the existing local knowledge stored on a computer right before a maybe imprecise query is sent. Users searching in an unknown problem domain can therefore expect more accurate search results. But even experts in an area can get benefit from this, as their queries will likely not be expanded with improper terms, when using documents of their area of expertise. Even queries containing homonyms can be properly expanded as well. Another advantage is the narrowed amount of search results of an expanded query. The technical implementation is based on a Firefox browser extension, called "FXResearcher", which is composed of a full text indexer and a module for query expansion.

Keywords: query expansion, browser extension, search engine, local text corpora, full text indexing, information retrieval, Firefox, P2P, Java

1. Motivation and Introduction

Private computers store a huge amount of data such as text documents. This knowledge is usually hidden from web search engines and a user can only use their contents with own efforts when conducting an online search. Search engines like Google [1] try to help their users in narrowing the amount of search results by offering suggestions to expand queries with terms that often occur with the requested terms in documents or based on frequently entered additional terms. Hence the search context for a query is created on the server side. But the local knowledge on a client computer might be sufficient and maybe more adequate to build proper queries right before the search request is sent. Under the assumption that the requesting user's computer stores documents matching a query's topic which contain terms that are likely candidates for query expansion, this new approach for client-sided query generation could result in more relevant search results matching the interests of the requesting user. This paper addresses this approach. We herein introduce a system that is capable of automatic query expansion through the usage of local text documents, while it is embedded in a Firefox web browser extension. This extension is called "FXResearcher" (short for "Firefox Researcher") and is composed of a full text indexer and a module for query expansion, called "Researcher". The basic workflow is as follows: It takes a user's query, tries to find matching documents on the local computer, calculates terms to expand this query based on a selected subset of the matching documents and sends the expanded queries to a web search engine like Yahoo! [2] or Google [1]. We show that query expansion performed on the user's computer can bring up promising search results. This way the building of search queries is directly influenced by the local knowledge and the search context is created before the request is sent to the web search engine. In the next chapter we will give a short introduction to query expansion in general before we will describe the functionality of the "Researcher". Chapter four deals with the architecture of "FXResearcher" and the reasons why we chose a browser extension to implement the above mentioned functions. We will also describe how the integrated modules work together. Chapter five provides some results of conducted tests with "FXResearcher" before chapter six gives a view on how it will be extended with a peer-to-peer component in the future.

2. Query Expansion

The driving force behind the development of query expansion techniques was to find solutions to the vocabulary mismatch problem. A search for e.g. "notebook" should also return results for its synonym "laptop". Therefore the aim is to expand queries with their synonyms and other related words in order to receive more relevant results. For instance a related word to the query "crocodile" might be "alligator". To find such query expansion terms several techniques have been developed in the recent years, of which some will be mentioned here. As an example, the often used relevance feedback strategy by Rocchio [3] adds terms of relevant documents to a query and subtracts terms of non-relevant documents. The decision whether a document is relevant or not is made by the user (Rocchio's method). If this relevance is automatically calculated without user intervention we speak of pseudo relevance feedback. The Vector Space Model (VSM) is often used to automatically determine the rank of documents by calculating the deviation of angles between the documents' term vectors and the query vector and is well suited for automatic relevance feedback [4]. A possible approach to expand a query is to select terms with the highest weight of the most similar documents, in case term weighting has been performed, e.g. using the TF-IDF model [5]. Another method based on the VSM is Latent Semantic Indexing (LSI) [6] which can be used to automatically extract concepts of document collections by mapping synonymous terms to the same dimension. It reduces the dimensionality of the original term-document-matrix. A problem is of course the interpretation of the gained dimensions. Queries can therefore be expanded with terms of the found concepts in order to get more relevant search results. Especially spreading activation networks [7][8] as models for connectionist information retrieval are well suited for query expansion because term expansion is one of their inherent properties. Related and connected terms can be easily found with this technique. For instance, given a term-document-matrix (like in the VSM) a bipartite graph of terms and document nodes with weighted edges is built. Then the node of the query (when found) is activated and its energy is distributed among the edges of connected nodes. Nodes with an activation level greater than a specific threshold are related to the initial query. In our approach we combine a relevance feedback mechanism to assist the user in selecting proper documents for query expansion with a spreading activation algorithm that returns highly activated terms in the selected document collection which are then used as expansion terms. Next we will describe how the "Researcher", our basic implementation for query expansion, works and explain the relevance feedback mechanism afterwards.

3. The Researcher

The task of the "Researcher"-component in "FXResearcher" is to generate expansion terms to a given query Q by algorithmically selecting those terms out of a local document set D1 in order to build expanded queries. These queries are then sent to a web search engine. Optionally, the 10 best results for each expanded query will be automatically downloaded and ranked according to D1. This module was originally designed and written as a standalone solution based on Java. It was then modified to work in a Firefox extension to increase the usability and provide a greater interactivity during the search process. The "Researcher" was conceived in the project "Search for text documents in large distributed systems" supported by Deutsche Forschungsgemeinschaft (DFG). It is working for German and English text documents. The applied software components for semantic text analysis were provided by the NLP department of the University of Leipzig [9].

Now we want to describe in detail, how the implemented query expansion method works.

- At first the user enters a query Q and gives a document set D1 for query expansion.
- Then a profile is calculated using D1 by concatenating all texts in D1 to an aggregation text T and extracting the most significant k words of T. The significance is determined through the usage of a well-balanced reference corpus. With this corpus it is possible to estimate the frequency of words in everyday-language. The usage of such sample collections was proposed by [10] and [11]. We use a likelihood ratio measure to compare the relative frequency of a term in T to its relative frequency in the reference corpus because of its proper mathematical foundation [12]. Terms whose frequency in T significantly surpasses the one estimated from the reference corpus, will be ranked highly.
- The aggregation text T is further used to calculate significant co-occurrences for each of the k words in the profile. These co-occurrences are saved in the co-occurrence graph G using a matrix.
- A spreading activation algorithm is then performed on this graph, which leads to an activation level and ranking for each of the k words in the profile related to the given query Q, whose elements are highly activated at the beginning and at the end. The n highest ranked words in the profile are then considered the expansion set E.
- Now the expanded queries (Q1…Qm) are generated by randomly determining the length of the current query Qi and then using the activation levels (as a probability distribution) to determine the positions of the selected elements of E.
- The expanded queries are now sent to Yahoo! [2] or Google [1].
- The “Researcher” then optionally downloads and ranks the 10 best results for each query while avoiding duplicates among the newly found documents. This is called the document set D2. To determine the ranking of the documents in D2, the profile and the terms of the elements in D2 are considered as vectors and their cosine similarity is calculated.

If it occurs that the initial query Q can not be expanded with the document set D1 it might be the reason that the topics in D1 differ too much among the documents or from the query itself. In this case only the initial query Q is sent. Therefore it is sensible to select documents with topics matching the query Q, as documents with homogeneous topics will likely result in more suitable expansion terms. How “FXResearcher” helps the user to select those documents as a relevance feedback will be discussed in the next section.

4. The Firefox Extension “FXResearcher”

After the first version of the “Researcher” had been developed and brought up useful results in expanding query terms, we wanted to give it a graphical interface that would be easily usable. Therefore we decided to integrate the “Researcher” into a browser extension for Firefox, as this web browser offers many powerful application programming interfaces (APIs), e.g. to build rich graphical user interfaces (GUI). It is even possible to run Java code from within an extension and to conveniently present search results on a webpage. We called this extension “FXResearcher”. The user should not notice a big difference from the known browsing experience when using this extension, as it can be controlled via a local webpage that is delivered with “FXResearcher”. The query results are also shown on this page.

4.1 The Architecture of “FXResearcher”

As mentioned above, the “Researcher” uses a set of documents to build query expansion terms. The selection of proper documents for this purpose is an immanent problem. The documents should semantically match a query in order to produce expansion terms so that the expanded search will bring up acceptable results. Therefore we decided to integrate a desktop search (full text indexer by the Leipzig NLP department [9]) into the extension beside the query expansion module.

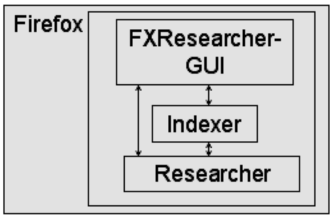


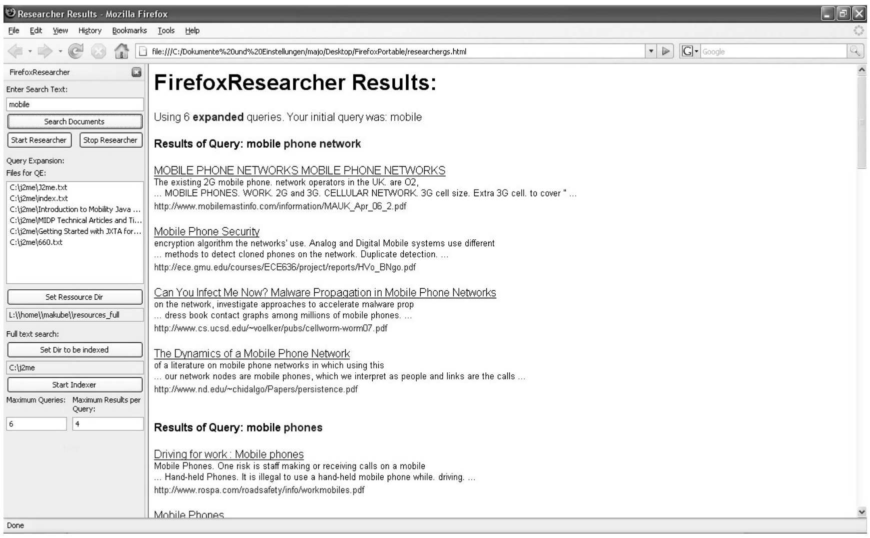
Figure 1. Architecture of “FXResearcher”

The indexer has the purpose to recursively scan specified local directories for documents and to incrementally index them as they are found. These documents can then be selected by the user to be used for query expansion. To select only proper documents, the user is given a small portion of the text around the matching terms for each found document. This way the user can easily judge a document’s relevance to a query. As a relevance feedback this solution promises to deliver topically homogeneous expansion terms during the query expansion process. The gained expanded queries can be further reviewed by the user before they are sent to the web search engine. At the time of writing this paper the web search results of the expanded queries are provided by the Yahoo! web search API [13]. Figure 1 shows the architecture of "FXResearcher".

4.2 User Guidance

A search with "FXResearcher" is conducted in these main steps:

- The user starts a search for local documents, giving an initial query
- The indexer responds with a list of documents matching this query
- The user selects some of these documents for query expansion and starts the "Researcher" (query expansion process)
- The user selects some of the suggested expanded queries and sends them to a web search engine
- The results of the expanded queries are presented on a webpage (fig. 2)



Researcher Results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Dokumente%20und%20Einstellungen/majo/Desktop/FirefoxPortable/researchergs.html
Google
FirefoxResearcher
Search Documents
Start Researcher
Stop Researcher
Set Ressource Dir
Set Dir to be indexed
Start Indexer
Done

Enter Search Text: mobile

Query Expansion:
Files for QE:
C:\j2me\J2me.txt
C:\j2me\index.txt
C:\j2me\Introduction to Mobility Java ...
C:\j2me\MIDP Technical Articles and Ti...
C:\j2me\Getting Started with JXTA for...
C:\j2me\660.txt

L:\\home\\makube\\resources_full

Full text search:
C:\j2me

Maximum Queries: 6
Maximum Results per Query: 4

FirefoxResearcher Results:

Using 6 **expanded** queries. Your initial query was: mobile

Results of Query: mobile phone network

MOBILE PHONE NETWORKS MOBILE PHONE NETWORKS
The existing 2G mobile phone. network operators in the UK. are O2,
... MOBILE PHONES. WORK. 2G and 3G. CELLULAR NETWORK. 3G cell size. Extra 3G cell. to cover " ...
http://www.mobilemastinfo.com/information/MAUK_Apr_06_2.pdf

Mobile Phone Security
encryption algorithm the networks' use. Analog and Digital Mobile systems use different
... methods to detect cloned phones on the network. Duplicate detection. ...
http://ece.gmu.edu/courses/ECE636/project/reports/HVo_BNgo.pdf

Can You Infect Me Now? Malware Propagation in Mobile Phone Networks
on the network, investigate approaches to accelerate malware prop
... dress book contact graphs among millions of mobile phones. ...
http://www.cs.ucsd.edu/~voelker/pubs/cellworm-worm07.pdf

The Dynamics of a Mobile Phone Network
of a literature on mobile phone networks in which using this
... our network nodes are mobile phones, which we interpret as people and links are the calls ...
http://www.nd.edu/~chidalgo/Papers/persistence.pdf

Results of Query: mobile phones

Driving for work : Mobile phones
Mobile Phones. One risk is staff making or receiving calls on a mobile
... Hand-held Phones. It is illegal to use a hand-held mobile phone while. driving. ...
http://www.rospa.com/roadsafety/info/workmobiles.pdf

Mobile Phones

Figure 2. Web Search Results for Expanded Queries

It is even possible for the user's convenience to automate the process of selecting local documents and submitting expanded queries to the web search engine, so that step three and four can be carried out without user intervention. For this purpose the user can specify a maximum number of local results (documents) that should be automatically used for query expansion and a maximum number of expanded queries to be sent to the web search engine.

4.3 Preserving Privacy

Admittedly, the last-mentioned feature can arise privacy issues as the automatically selected documents may contain confidential or private data such as bank account information that might become part of expanded queries. Therefore it is sensible to choose a separate folder with noncritical documents to be indexed. Of course it should be avoided to give the indexer access to the root directory of the computer.

5. Test Scenarios

In our experimental setup we chose several queries of different topics and let the "Researcher" expand these queries with a set of found local documents by the aforementioned indexer. A maximum of 10 expanded requests and a maximum of 10 results per request were allowed. We used the web search engine Yahoo! [13] to send the expanded queries to. Next we will show the results of three conducted tests with the "Researcher" in comparison to the suggestions for related queries by Google and Yahoo!:

- Test 1: Topic „P2P und Semantic Web"
  Query: "Semantic Web"

  Results of: "Researcher"
  - Given: 7 PDF-documents found with Yahoo!
  - Found query expansion terms:
    Peer-to-Peer, Peer-to-Peer Distributed,
    Peer-to-Peer Piazza, Peer-to-Peer Services,
    Services, Smart Services, Piazza Services
  - Additionally found documents matching the topic: 33

  Suggested related terms by "Google Suggest":
  - wikipedia, wiki, webbing, ontology, company, owl, scientific american, agents, business, gang

  Suggested related terms by "Yahoo! Suggest":
  - services, folksonomy ontology, xml, find similar music, berners lee, concept, expert
- Test 2: Topic "Affiliate Marketing"
  Query: "affiliate"

  Results of "Researcher":

  - Given: 6 PDF-documents found with Yahoo!
  - Found query expansion terms:
    marketing advertisers, advertisers,
    advertisers network, marketing online,
    network fees, marketing, network marketing,
    network, advertisers publishers, network advertisers
  - Additionally found documents matching the topic: 61

  Suggested related terms by "Google Suggest":
  - sec, forum, meaning, travel, clickbank, subsidiary, marketing, commission junction, window, summit, elite, engineers, computer services, future, managers group

  Suggested related terms by "Yahoo! Suggest":
  - marketing, programs, become a poker affiliate, internet poker, affiliatesellers, affiliates

- Test 3: Topic "Christmas Tree (Oil Wellhead)"
  Query: "christmas tree"
  Results of "Researcher":
  - Given: 3 PDF-documents found with Yahoo!
  - Found query expansion terms:
    oil platform, production, oil well, oil production, well, well production, oil view, oil gas, subsea platform, delivery
  - Additionally found documents matching the topic: 42

  Suggested related terms by "Google Suggest":
  - Shop, farm, shop locations, hill, shop ma, farm kent, shop ct, farming, shop orange ct, artificial, pictures, sale, xmas

  Suggested related terms by "Yahoo! Suggest":
  - e-cards, index of under the, spode, decorations, dishes, under the, screen saver

With these and further tests we could prove that the "Researcher" produces usable expansion terms when documents with homogeneous topics are provided for query expansion. This is also the case when only a few local documents matching a specific query can be found. At the same time we could show that the quality of the offered related terms by Google and Yahoo! are not always satisfying. Even so, Google's suggestions are generally more suitable than those by Yahoo!, although also Google did not always manage to provide expansion terms for queries with different meanings. For instance, the query "christmas tree" in the meaning of an oil wellhead was never reflected in the list of related terms. Therefore, in direct comparison with both well-known services the "Researcher" provided the most suitable expanded queries and search results.

6. Conclusion

In this paper we introduced the Firefox browser extension “FXResearcher”, whose aim it is to support the user in searching for documents on the local computer and on web presences. Its new approach of client-sided query expansion based on local text documents in order to find more proper documents in the web has been elaborated on in detail. We also pointed out that these documents should match a query’s topic, because documents with homogeneous topics contribute to the calculation of proper expansion terms for a query. To find these matching documents “FXResearcher” integrates a full text indexer that incrementally indexes specified folders on the local computer. Documents in this index can then be selected for query expansion as a relevance feedback when they contain terms of a search query. This approach is promising, because a query will likely not be expanded with improper terms as it still occurs when query expansion will be first performed by the requested web search engine. The latter solution might completely fail when for instance an expert searches for documents with a quite specific term that does not occur in many documents. The usage of suggested improper expansion terms could return only a few and inadequate results. The consideration of the local knowledge for query expansion on the other hand can provide the expert with proper expansion terms on his area of expertise and therefore return more expected search results. With this approach even queries consisting of homonyms can be properly expanded. In a future version of “FXResearcher” a peer-to-peer (P2P) module will be integrated. This module will allow users to semantically search for documents on participating peers. The advantages of a P2P-system like scalability and availability will also support this function. Many researchers active in the field of P2P information retrieval recognized this and started the development of search technologies like those described in [10], [11], [14] and [15]. Our aim is to provide techniques to route queries to semantically matching peers by analyzing their peer profiles that consist of compact representations of their locally offered documents. The usage of peer descriptions in form of profiles will even avoid blind search methods like those applied in Gnutella [16]. This leads to a clustering of semantically similar peers, forming a small world network structure [17]. Even in this scenario query expansion can significantly enhance the quality of the result set. For instance, incoming queries can be analyzed and expanded on a peer using the local knowledge in the same manner as described above. The retrieved documents will then be added to the result set and an expanded query can be forwarded to peers with similar peer profiles. Additionally, peers that have contributed many documents could also add terms from their profile to the query. This approach is quite useful as no global knowledge of all participating peers has to be computed and maintained. Another advantage of using query expansion is the compensation of the missing completeness of peer profiles due to the fact that a peer profile has to be compact in order to be attached to messages and stored in other peers' routing tables. With this additional module "FXResearcher" will be turned into a multifunctional browser-based semantic search tool that addresses the interactive search for documents on the local computer, the World Wide Web and on remote computers through P2P information retrieval.

Acknowledgements

We would like to explicitly thank Dr. David Huynh of the Massachusetts Institute of Technology (MIT) for his great work to support Java code in Firefox extensions. His project "JavaFirefoxExtension" [18] made it possible for us to easily integrate the functionality of the "Researcher" into a Firefox extension. His helpfulness regarding the solution of special problems in the implementation was very much appreciated.

References

[1] WEBSITE OF GOOGLE, *www.google.com.*

[2] WEBSITE OF YAHOO!, *www.yahoo.com.*

[3] ROCCHIO J., *Relevance Feedback in Information Retrieval*, The SMART Retrieval System: Experiments in Automatic Document Processing, chapter 14, pages 313-323, Prentice Hall, Inc., 1971.

[4] JÉRIBI L., RUMPLER B., Instance Cooperative Memory to Improve Query Expansion in Information Retrieval Systems, J. UCS 8(6): 591-601, 2002.

[5] SALTON G. et al., A vector space model for information retrieval, Journal of the American Society for Information Science, 18(11):613-620, 1975.

[6] DEERWESTER S. C. et al, Indexing by latent semantic analysis, Journal of the American Society of Information Science, 41(6):391–407, 1990.

[7] COLLINS A., LOFTUS E., A spreading activation theory of semantic processing, The Psychological Review, 82(6):407-428, 1975.

[8] PREECE, S., A spreading activation network model for information retrieval, PhD thesis, CS Dept., Univ. of Illinois, Urbana, IL, 1981.

[9] Website of the natural language processing group (NLP), University of Leipzig, *http://www.asv.informatik.uni-leipzig.de/opencms/asv/de/index.html.*

[10] TANG C. et al., pSearch: Information retrieval in structured overlays, ACM SIGCOMM Computer Communication Review, pages 89–94, 2003.

[11] KRONFOL A.Z., *FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine*, Princeton University, 2002.

[12] DUNNING T., *Accurate methods for the statistics of surprise and coincidence*, Computational Linguistics, 19(1):61–74, 1994.

[13] Website of the Yahoo! Search Web Services, *http://developer.yahoo.com/search/web/V1/webSearch.html.*

[14] CUENCA-ACUNA F. M. et al., *PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities*, 12th International Symposium on High Performance Distributed Computing (HPDC), 2003.

[15] KALOGERAKI V. et al., *A Local Search Mechanism for Peer-to-Peer Networks*, In Proceedings of the Eleventh International Conference on Information and Knowledge Management, pages 300–307, 2002.

[16] Website of Gnutella, *www.gnutella.com.*

[17] KLEINBERG J., *The Small-World Phenomenon: An Algorithmic Perspective*, In Proceedings of the 32nd ACM Symposium on Theory of Computing, 2000.

[18] Website Of the Project “JavaFirefoxExtension”, *http://simile.mit.edu/wiki/Java_Firefox_Extension.*