IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Context Pseudo Relevance Feedback using Machine Learning Technique

Dr. Jebamalar Tamilselvi. J

*Associate Professor,*
*Department of Computer Science,*
*SRM Institute of Science and Technology,*
*Ramapuram, Chennai*

Dr. V. Umarani

*Assistant Professor,*
*Department of Computer Science and Engineering ,*
*Saveetha Engineering College,*
*Thandalam, Chennai*

***Abstract*** - Nowadays the number of users has increased in use of internet web access as well as the growth of data has also increased. This can be a hassle for users to find exactly relevant information from the Internet relatively quickly. The search process on the web is also inaccurate and it takes a lot of time to get results. Domain-specific web search engines contain information that is specific to the subject at hand. This domain-specific web search engine aims to improve accuracy and provide additional functionality. This makes it easy to connect young minds with start-ups that have turned out to be quite different from common web search engines. This proposed task uses a focused crawler that attempts to index only web pages that contain information about jobs, launch-related events, and news. It also uses contextual pseudo-relevance feedback using machine learning algorithms to get more relevant documents than regular search engines. This proposed task produces search results by reflecting feedback without human intervention. This task uses machine learning techniques to improve accuracy with domain-specific web search engines for better results.

***Index Terms –Web Crawler, Inverted Indexes, Relevance Feedback, Ranking***

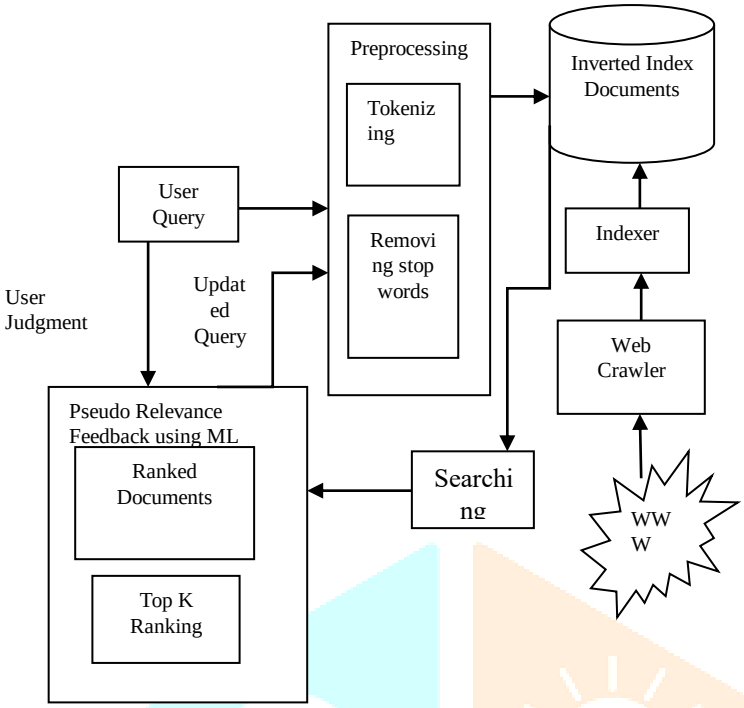
## I. INTRODUCTION

This Domain-specific search engines are designed to provide more accurate and better search results for job and startup related information, helping users find the most relevant search results easily. This is quite different from a typical web search engine. As a general trend, you can see that many short queries are being made to search engines. In this way, the user's information needs are provided with some meaningful search terms. One way to improve the effectiveness of polling is to consider ways to automatically improve the requests sent. To specify some important search terms and generate effective search results, the system uses search engine optimization techniques to perform query extensions with pseudo-relevance feedback. Pseudo-relevance feedback (PRF) is often seen as an effective way to amplify search queries. The first ranked documents retrieved from the system are considered relevant and are used as implicit feedback to generate additional terminology. By using content target pseudo-relevance feedback, you get more documents as search results than you would with a regular search engine because you use query extensions. It also has a built-in PageRank implementation that can be turned on and off from the application's UI. The goal of this system is to develop a domain-specific web search engine that focuses primarily on information about jobs and start-ups. The system focuses on achieving search engine optimization by extending the query without the user having to specify all of the exact keywords in the search query. Query extensions help generate more search results than regular search engines. Search results generated by query extensions display the most relevant documents at the top, but simple search engines rarely display relevant documents at the top. The system aims to provide search results in the most relevant order by ranking documents using the PageRank algorithm. This paper uses inverted indexes, PRF, and Pagerank algorithms to get better results than the existing work of domain-specific search engines.

## II. ARCHITECTURAL DESIGN

The web crawler collects web pages from World Wide Web by crawling, which contains information about job opportunities and start-up company related events by carefully prioritizing the crawl frontier and managing the hyperlink exploration process. The crawled web pages has to be pre-processed in order to perform indexing. In pre-processing, tokenization process is done on web pages and the tokens are stemmed using the PorterStemmer by nltk, the stopwords are eliminated using the list of stop-words provided in the file stopwords.txt, the words shorter than 3 letters are not considered. Therefore, the inverted index is built and the tf-idf of each word-doc pair is computed and stored in the inverted index. The user gives a query and can enable or disable page Rank and pseudo Relevance feedback. If the pseudo relevance feedback is turned on, then the query is expanded and the query parser translates the search string given by the user into specific instructions for the search engine. After retrieving, the relevant documents are ranked based on its relevancy to the user query using PageRank Algorithm. PageRank works by counting the number and quality of links to a page to determine

a rough estimate of how important the website is. Finally, the ranked documents are produced as results to the user.




*A. CRAWLER*

The crawler collects web pages that contain information about job opportunities and start-up company related events. The Initial list of URLs contained in the crawler frontier are known as seeds. The web crawler will constantly ask the frontier what pages to visit. As the crawler visits each of those pages, it will inform the frontier with a response of each page. The crawler will also update the crawler frontier with any new hyperlink contained in those pages it has visited. These hyperlinks are added to the frontier and will visit those new web pages based on the policies of crawler frontier. This process continues recursively until all URLs in the crawl frontier are visited.

The crawling process involves crawling the web pages from the World Wide Web based on the given seed url. The crawler works based on the following steps:

- Crawling happens by using the Queue module to access the resources in a synchronized way.
- Crawling happens with a breadth-first strategy, every page is dequeued, downloaded and parsed using the HTMLParser library, its links are extracted and checked to belong to the job domain, then added to the FIFO queue if it is in an appropriate format.
- The urls are also modified after they are extracted, every initial http becomes an https, all the slashed at the end are eliminated, the query strings are removed and also the intra-page links expressed by the hash symbol (#) are also removed in order not to let the crawler believe that more pages with a different inpage-links are different urls and so different pages.

*B. PRE-PROCESSING AND INDEXING*

Once crawling is done, the crawled web pages need to be pre-processed so that it can be indexed. Pre-processing of web pages involves tokenization, stemming and removal of stop words. After pre-processing, the indexing process is done by building an inverted index with the pre-processed words. TF-IDF weighting scheme is used to build inverted index. Indexing process collects, parses and stores data for use by the search engine. An Inverted index compiles database of text elements. The Inverted index data structure stores a mapping from content such as words or numbers, to its location in a document or a set of documents. It is a hash map like data structure that directs you from a word to a document or a web page. The following are the steps to build an inverted index. They are fetching the documents, Stemming of Root Word and Record Documents IDS.

The TF*IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the web, which is referred to as corpus. For a term **t** in a document **d**, the weight **Wt,d** of term t in document d is given by:

$Wt,d = TFt,d \log (N/DFt)$

Where:

- TFt,d is the number of occurrences of t in document d.
- DFt is the number of documents containing the term t.
- N is the total number of documents in the corpus.

*C. CONTEXT PSEUDO-RELEVANCE FEEDBACK*

When the user gives a query, the query is expanded using pseudo relevance feedback. This user feedback will be automated using machine learning algorithm to reduce the human interaction. This method does normal retrieval to find an initial set of most relevant documents, to it assume that the top "k" ranked documents are relevant, and finally to do relevance feedback. It takes the results returned by initial query as relevant and Selects top 20-30 terms from these documents using for instance tf-idf weights and does Query Expansion, add these terms to query, and then match the returned documents for this query and finally return the most relevant documents.

Pseudo relevance feedback, also known as blind relevance feedback, provides a method for automatic local analysis. It automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction. The method is to do normal retrieval to find an initial set of most relevant documents, then assume that the top "k" ranked documents are relevant, and finally to do relevance feedback as before under this assumption.

The procedure is:

1. Take the results returned by the initial query as relevant results (only top k with k being between 10 and 50 in most experiments).
2. Select top 20-30 (indicative number) terms from these documents using for instance tf-idf weights.
3. Do Query Expansion, add these terms to query, and then match the returned documents for this query and finally return the most relevant documents.

A simple static query expansion based on synonyms would have been too simple and would not have been able to capture contents that are related but have a different semantic. However, the starting point always has to be the user's query, as there is nothing else except that initially explicit relevance feedback when the user is asked which other words he would want to include in his query would probably be better, but at the same time it could be annoying for the user to respond to

some questions while searching. A pseudo-relevance feedback seems to be a more appropriate and easy way to run in the background some more complex query that the user is not even aware of. To extract some words that could express the context of the formulated query we use some intrinsic information that the initially retrieved documents have.

*D. RANKING*

After retrieving, the relevant documents are ranked based on its relevancy to the user query using PageRank Algorithm. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. PageRank assigns a numerical weighting to each element of a hyperlink set of documents, such as World Wide Web, with the purpose of "measuring" its relative importance within the set. During the pre-processing of pages, the links are extracted and based on the link connections a world graph is created. The implementation of Page Rank was such that it created a strongly-connected-component with the entire graph, which means that from every node it is possible to get to another node of the graph with a non-null probability. With this interpretation there is no possibility that a random walker would get stuck in a page. The page ranks were a bit difficult to integrate into the scoring of documents to rank them.

## III. RESULT

The spider starts crawling the web from the seed URL and finds all the links on each of the page that it visits and stores it in the URL list. This process of crawling continues till there are no further links to visit. The crawled URLs are stored in a specified directory for indexing. After preprocessing the keywords in the expanded query are matched with the indexed documents and the relevant documents are fetched and displayed in a relevance order based on the keyword quality score computed using PageRank Algorithm.The results are compared with the traditional web crawler and focused web crawler with PRF. As the result shows, Focused crawler with PRF yields accurate documents and decreased the fetching time of the documents.

TABLE I
NUMBER OF LINKS OF COLLECTED

| Type | Traditional Web Crawler | Keyword Focused Web Crawler | Traditional Web Crawler with PRF | Keyword Focused Web Crawler with PRF | Keyword Focused Web Crawler with PRF using ML |
|---|---|---|---|---|---|
| **Total of Extracted Links** | 1240 | 630 | 432 | 249 | 203 |
| **Relevant number of links** | 964 | 467 | 259 | 230 | 190 |
| **Crawling Time** | 660 Sec | 220 Sec | 350 Sec | 200 Sec | 180 Sec |

## IV. ADVANTAGES

This search engine has the ability to extend the set of retrieved documents. In fact, if there are few documents that contain any of the words in the query, a simple search engine will only retrieve those few documents. If you use intelligent PRF components instead, the result set can be much larger and the size of the retrieved set can exceed 100 documents. The system is designed to crawl only web pages that contain startups and job listings, so the results are more relevant and accurate than the user's expectations. This user expectation is supported by machine learning techniques to improve the accuracy of the results. It actually finds what the user is looking for as the first result, but simple search engines don't even find it in the top 10 results. It focuses only on specific domains, resulting in faster search results.

## V. CONCLUSION

In this paper, we aimed to explore the possibility of improving search efficiency by query extension when considering only a specific search range. Machine learning technology is used to automate user feedback through enhancements. Our experimental results show that this domain search engine returns the most relevant documents as search results and responds faster than regular search engines because they are limited to a particular domain. This paper suggests some interesting research tools for our future research. Further research is needed to explore the broader query characteristics and possible technical improvements. For ambiguous queries, the final search will be more efficient if the decontamination process can improve accuracy. With the supervised term selection method, the results achieved are not satisfactory in terms of accuracy.

## REFERENCES

[1] Marijn Koolen, Toine Bogers, Maria Gäde, Mark Hall, Iris Hendrickx, Hugo Huurdeman, Jaap Kamps, Mette Skov, Suzan Verberne, and David Walsh. 2016. *"Overview of the CLEF 2016 Social Book Search Lab. In Experimental IR Meets Multilinguality, Multimodality, and Interaction".* 351–370.

[2] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. *"Image-based recommendations on styles and substitutes".* In SIGIR. 43–52.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013.*"Efficient estimation of word representations in vector space"*. ICLR Workshop (2013).

[4] Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M Jose. 2011. *"Handling data sparsity in collaborative filtering using emotion and semantic based features"*. In SIGIR. 625–634.

[5] Cataldo Musto, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2019. *"Justifying recommendations through aspect-based sentiment analysis of users reviews".* In UMAP. 4–12.

[6] Joseph John Rocchio. 1971. *"Relevance feedback in information retrieval. The SMART retrieval system: experiments in automatic document processing",* 313–323.

[7] Peter D Turney and Michael L Littman. 2003. "*Measuring praise and criticism: Inference of semantic orientation from association"*. TOIS 21, 4 (2003), 315–346.

[8] Heng-Li Yang and August FY Chao. 2018. "*Sentiment annotations for reviews: an information quality perspective"*. Online Information Review 42, 5 (2018), 579–594.

[9] Zheng Ye and Jimmy Xiangji Huang. 2014. *"A simple term frequency transformation model for effective pseudo relevance feedback*". In SIGIR. 323–332.

[10]. Yuan Zhang, Dong Wang, Yan Zhang. *"Neural IR Meets Graph Embedding: A Ranking Model for Product Search"*. The Web Conference, San Francisco, CA, USA. ACM ISBN 123-4567-24-567, May2019.

[11] Navjot Kaur, Himanshu Aggarwal, *"Query reformulation approach using domain specific ontology for semantic information retrieval".* International Journal of Information Technology,Springer.https://doi.org/10.1007/ s41870- 020-00464-2.2020.

[12] Ping Wang, Tian Shi, Chandan K. Reddy. "*Text-toSQL Generation for Question An- swering on Electronic Medical Records".*IW3C2 (International World Wide Web Confer- ence Committee).,ACM ISBN978-1-4503-7023-3/20/04,2020.

[13] Wu Zetal*."Verbssemantics and lexical selection".*In:Proceedingsofthe32ndannual meeting on Association for Computational Linguistics. Association for Computational Linguistics.

[14] Keet Sugathadasa, Buddhi Ayeshaal*."Legal Document Retrieval using Document Vector Embeddings and Deep Learning".* Computing Conference 2018 10-12 July2018.

[15] Xu, B., Lin, H.,Lin,Y.(2016)."*Assessment of learning to rank methods for query expansion".* Journal of the Association for Information Science and Technology, 67(6): 1345-1357.2016.

[16] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria. *"Recent Trends in Deep Learning Based Natural Language Processing",* arXiv, IEEE Computational intelli- gence magazine,Nov 2018.

[17] T. Kawamura, K. Kozaki, T. Kushida, K. Watanabe, and K. Matsumura*, "Expanding science and technology thesauri from bibliographic datasets using word embedding,"* in Proc. IEEE Int. Conf. Tools Artif. Intell., Nov.2017.

[18] Shickel, B., Tighe, P. J., Bihorac, A, Rashidi *"Deep EHR: A Survey of Recent Ad- vances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis",* JBHI.2767063, IEEE, 2017.

[19] Jimmi, Guido Zuccon and Bevan Koopman*."Knowledge Graphs and Semantics in Text Analysis and Retrieval".* Information Retrieval Journal. https://doi.org/10.1007/s10791- 018-9344-z. Springer Nature B.V.2018.

[20] Sumit Kumar Mishra, V.K. Singh. *"Building Semantic Information Retrieval System For Legal Cases From Heterogeneous Adapted And Diverse Data Sources Using Extended GAIAMethodologyForMultiAgentSystem",* byIEEE.978-1-5090-6785-5/18/©2018.