Focused Crawling System based on Improved LSI

[1]Radhika Gupta, [2]AP Nidhi

Department of computer Science, Swami Vivekanand Institute of Engineering & Technology,
Punjab Technical University, Jalandhar, India

Abstract: ***In this research work we have developed a semi-deterministic algorithm and a scoring system that takes advantage of the Latent Semantic indexing scoring system for crawling web pages that belong to particular domain or is specific to the topic .The proposed algorithm calculates a preference factor in addition to the LSI score to determine which web page needs to preferred for crawling by the multi threaded crawler application, by doing this we were able to produce a retrieval system that has high recall and precision values as it builds a queue which is specific to a particular domain/topic which would not have been possible in Breath first and only LSI based information retrieval systems.***

Keywords: LSI, Breath first crawler, focused crawler

1. Introduction

Crawling [1] is a highly resource intensive task which requires coordination of multiple threads and large spectrum of bandwidth. Secondly, crawling is semi-undeterministic approach for indexing and getting information, therefore, it is a necessity to develop an algorithm which helps in saving computational resources and bandwidth [2], Hence, the need for focused crawlers. These focused crawlers may be domain specific or knowledge specific in nature, which helps to develop an information retrieval system which will have high precision and recall values due to the fact that it has crawled highly relevant pages. The challenge is to develop algorithm which work on the principal of calculating score on the basis of context of the knowledge domain on which we are working on and the websites which are being crawled. Latent Semantic Indexing (LSI)[3,4] is one of the promising models to do so, and there is an urgent need to develop a scoring system that can help to crawl pages that are specific to a particular domain therefore we proposed a crawling system that improvises on the latent semantic indexing scoring system[2,3,4] to argument those pages to crawl that can help to reduce resource consumptions due to its mathematical model ,our proposed system does not intend to work on the limitations but rather take advantage of LSI system and modified it in such a way that is calculator weightage for domain specific terms together pages related specific to it.

2. Proposed Work

The proposed system work on calculating performance factor for domain specific terms for searching, based on following mathematical expressions

1. $PF_{wt} = \frac{\#\text{ of relevant words found by LSI + textfact+relevamt words € domain}}{\#\text{ of words in domain specific dictionary}}$ ......Eq 1

2. $PF = \frac{a+b}{c}$ Eq 2

Where a=number of relevant words found by LSI Score cut off.

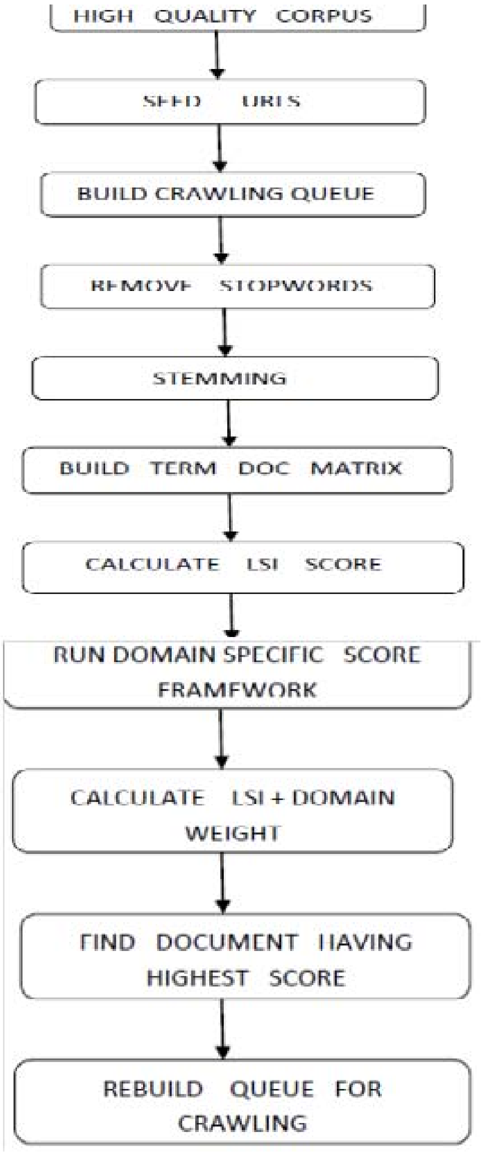
b= number of relevant words found that belongs to movie dictionary

c= total number of words in movie dictionary

mLSI = $\sum_{1}^{n=\#of\ terms\ in\ web\ doc}(LSI+PF)$

nLSI = final score expression

3. Implementation


HIGH QUALITY CORPUS
SEED URLS
BUILD CRAWLING QUEUE
REMOVE STOPWORDS
STEMMING
BUILD TERM DOC MATRIX
CALCULATE LSI SCORE
RUN DOMAIN SPECIFIC SCORE FRAMEWORK
CALCULATE LSI + DOMAIN WEIGHT
FIND DOCUMENT HAVING HIGHEST SCORE
REBUILD QUEUE FOR CRAWLING

1. The first step in our work is to select domain specific keywords, movies was chosen as our primary domain and A dictionary of all the key words (name of the movies, lyrics of the songs, abbreviations of the movies like DDLJ etc.) relating to movies is build. The movie dictionary [focus] consists of more than 10,000 elements in it [2].
2. The crawler is initially supplied with seed URLs which are specific to the movie domain [2].
3. In the next step is tokenizing. All files in the corpus are decomposed into tokens and the stop words are removed from those tokens [2].
4. Now we have with us a list of terms which are free from stop words and any numerical values or arbitrary symbol of irrelevance [2].
5. In the next step a term document matrix is created. In order to build the LSA also known as known as Latent Semantic Indexing (LSI) model we need to create Term Document Matrix (TDM)[5] from the corpus which is then passed to Single Value Decomposition (SVD)[6] to obtain U matrix ,S matrix and V matrix values. In our Term-Document matrix the columns correspond to the documents and the rows correspond to the terms. Each entry in the Term-Document matrix corresponds to the number of times a particular word is used in a particular document also known as the frequency of the term [2].
6. Once we have built our TDM matrix [5], we now use a Singular Value Decomposition or SVD to analyze the matrix for us. The use of SVD is that, it figures out how many dimensions or "concepts" to use when approximating the matrix. When we have supply TDM to SVD [6] we will get three matrix name U, S, V.
7. LSI score for each term is calculated and stored.
8. Once the LSI score for each term is calculated, then we calculate the total LSI score of a particular page which is equal to the sum of LSI values of each term appearing in that page and also calculate preference score or weigh [eq1] according to the LSI score of the documents these are arranged in sorted order [2].
9. Now, further crawling of the URLs obtained in step 8 takes place and the links obtained through the crawl path are extracted. The link text is further subjected to the LSI score calculation and again the link with highest LSI value is crawled and crawling of these links takes place in decreasing order and this process repeats.
10. The extracted links are crawled and performance is evaluated of the system stored.
11. Finally the performance is evaluated.

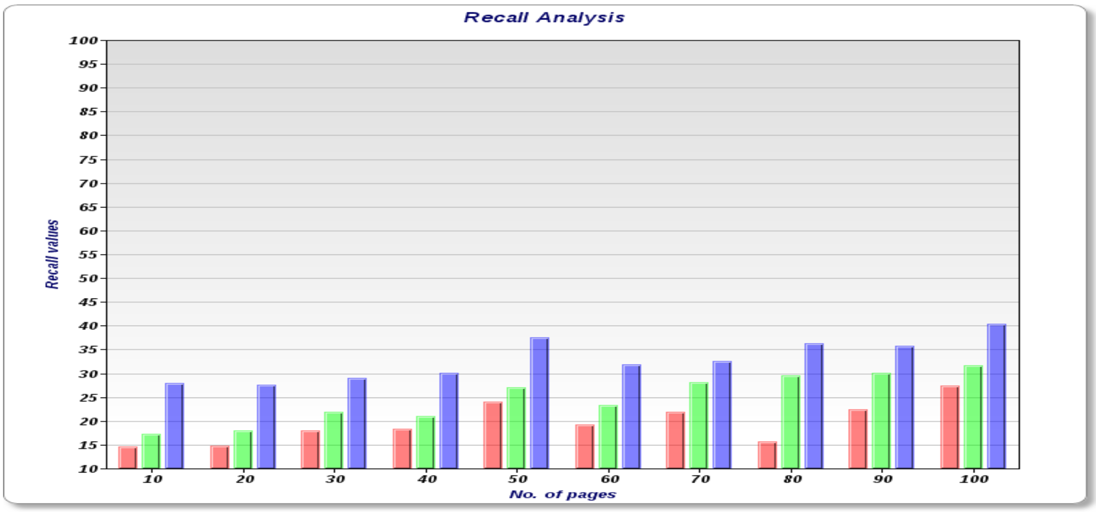
4. Results/Graphs

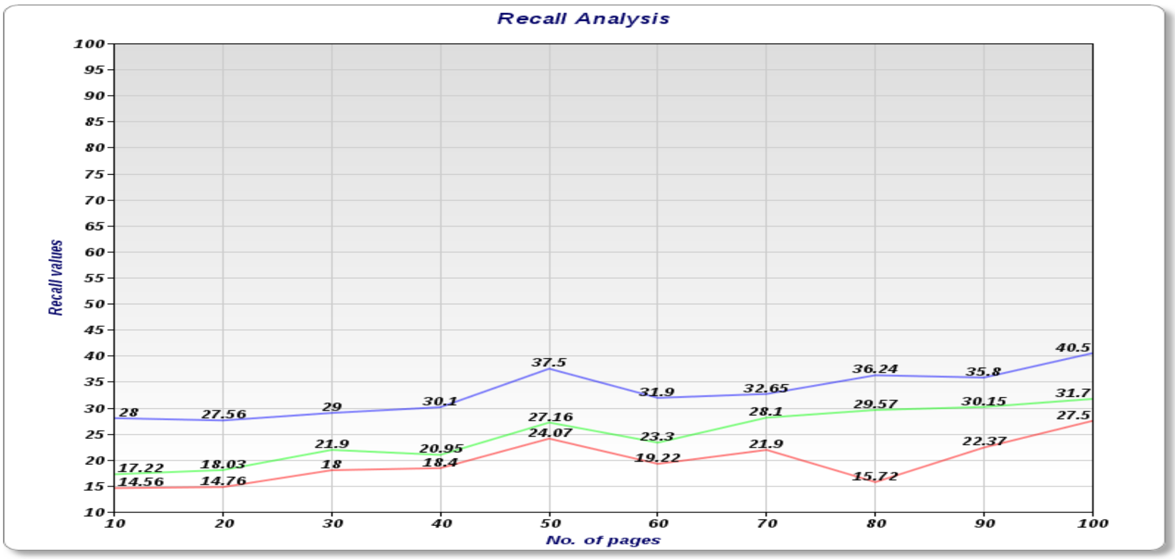
Algo 1:- Breadth first

Algo 2:- LSI

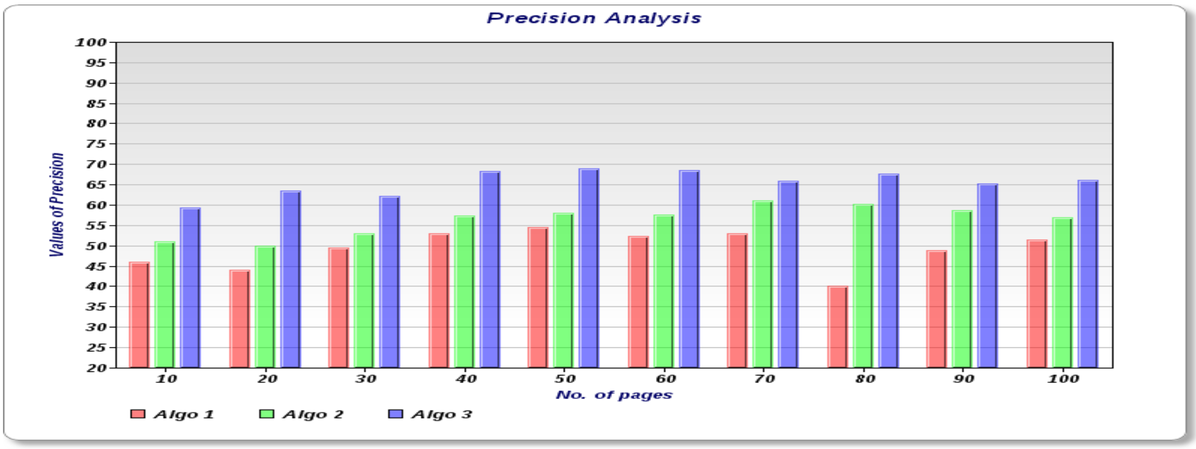
Algo 3:- Domain specific crawler.

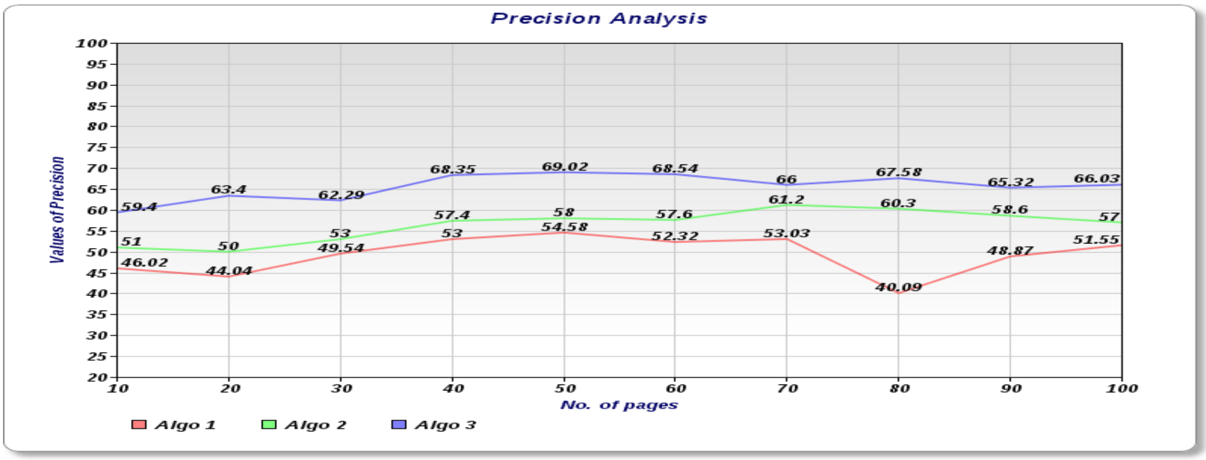
Recall Analysis





Precision Analysis





Interpretation of Graphs

Our focused crawler builds its corpus, which is specific to movies domain. Therefore, it is a model that works on the principle of selecting only those web documents for which as per Algo 3, it can gain information with respect to movie domain only. In this process it is intuitively reducing the uncertainty about the category of a document item being selected for crawling X provided by knowing the value of feature Y. Here item Y is the seed keywords or URLs or future hyperlinks or the titles.

Since the ultimate goal of Algo 3 or our focused movie crawler is to build a dataset that would provide a high information gain when used by a search engine or query engine, the selection of URLs and keywords is very important as it would lead to burning of less resources. As we are taking the advantage of highly optimized dictionary of terms related to movies (names of the movies, abbreviation of the movies, lyrics of songs, hero's name and heroine's name more than 10,000 unique dictionary elements) helps us in improving the recall and precision of our overall system.

It is apparent from the graph for recall analysis [8] that the recall value varies from 28% to 40.5% which reflects the completeness or sensitivity of our Algo 3. The recall value here means less number of crawl jobs that are false negative in nature , or in simple words , crawling less number of web documents that were selected erroneously or those web URLs which were supposed to be rejected but got selected in URL crawl priority queue.

It can also be seen from the precision[8] graph that precision values remains around 59.4% and 66.03% which is otherwise difficult to obtain had not the Algo 2 been implemented , because normally if recall value increases (in our case it is moderate) the precision often decreases as it gets harder to be precise when the sample space increases. But, in our result we can see that precision remains moderate, that means around 60% crawls are true positive in nature, or in simple words, the web documents which were supposed to be in priority queue were correctly selected.

5. Conclusion

In this thesis a domain specific movie crawler has been implemented. A domain specific crawler is useful for saving time and other resources since it is concerned with a particular domain. Hence we obtain highly relevant data which leads to high information gain and less resource wastage. Since, people today are keen on having information about movies so information about movies has been chosen as a domain to work on. Various methods of information retrieval have been studied and reviewed and based on this literature survey it was found that there is a requirement to build a crawler that takes into account the context of the words or phrases being searched for. LSI with preference factor mathematical model is one such promising model in the field of information retrieval. LSI uses a mathematical technique known as Singular Value Decomposition. This model has the ability to extract the conceptual content of a body of text by looking for relationships between the terms of the text. The evaluation of the work has been done by using the recall and precision values and it can be seen that the precision value and recall value following good rate to contribute to the accuracy 66.03% of the system.

6. Future Scope

These days many information retrieval systems are being created based on taxonomies, ontologies, knowledge bases. The users want information based on particular domains which would help them save time and effort and would help them retrieve more relevant and useful results. However there is still lot to do in the field of domain specific crawlers. Creation of more domain based crawlers is suggested in various areas such as chemistry, biology, medicine, etc. We can also add other machine learning algorithms like probabilistic algorithms, neural network etc which may result in even better precision.

7. Acknowledgement

I am thankful to A P Nidhi, Assistant Professor, Swami Vivekanand Institute of Engineering and Technology, Banur, for providing constant guidance and encouragement for this research work.

Reference

[1] http://en.wikipedia.org/wiki/Web_crawler.

[2] Radhika Gupta and AP Gurpinder Kaur, Review of Domain Based Crawling System, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.

[3] April Kontostathis a and William M. Pottenger b, A Framework for Understanding Latent Semantic Indexing (LSI) Performance, International journal on Information Processing and Management, Volume 42, January 2006 (Elsevier).

[4] Hong-Wei Hao1, Cui-Xia Mu, Xu-Cheng Yin, Shen Li, Zhi-Bin Wang,An Improved Topic Relevance Algorithm for Focused Crawling

[5] M. Berry, Z. Drmac, and E. Jessup. Matrices, vector spaces, and information retrieval. SIAM Review, 41:335–362, 1998.

[6] M.W. Berry and M. Brown. Understanding Search Engines. SIAM, Philadelphia, 1999.

[7] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6):391–407, 1990.

[8] http://www.creighton.edu/fileadmin/user/HSL/docs/ref/Searching__Recall_Precision.pdf

[9] Ritendra Datta , Dhiraj Joshi, Jia Li, James Z. Wang , " Image retrieval: Ideas, influences, and trends of the new age ", ACM Comput. Surv. Vol. 40, No. 2. , May 2008.