SuperPages: Simplified Search which Relates*

Muhammad Usman Akram†
University of Trento
Trento
muhammadusman.akram@studenti.unitn.it

ABSTRACT

With the introduction of web 2.0, growth of web accelerated as more people started generating and sharing content. Now, even most widely used encyclopedia, "Wikipedia" is a collection of use generated content. Along with huge number of user blogs, which share large amount of information. Thus, web has become enormously huge and even with excellent key words search engines, whose searches usually result in list of tens of thousands of pages but finding information among those pages is not possible by end user exploring them one by one. Key word search engines, return the pages with some ranked order, containing key words of interest, which were provided by user in query. User's query is key words but search engines output results to users without any semantical understanding of query or resulting pages. Current, keyword search engine cannot show what other topics might be similar or related to user's query. Tough search engines provide related searches done by other users, which might or might not be related to users query. User will have to read through result documents to find useful related concepts, while much of documents in results can form groups of similar or semantically related documents. Which can save user from issuing issuing new queries like "similar pages" or for other related concepts. For, example if user query is "Artificial Intelligence online" is user searching for some online projects? or some online courses? or just some online algorithms? A key word search engine will put all of them without any structural organization just based on there authority on keywords based on there interaction with other pages. What if we can present user with hieratically organized result? Like online projects, classes and algorithms, this will make search much user friendly and less time consuming, as it will reduce time to reach desired information. Instead of giving user a generalized list of result and waiting for user to issue more specific queries, there is need for a system, which understands relations between keywords of query and other related concepts and is able to present results in hieratically organized structure, which have similar or semantically related pages in a group or category. Our approach to this system is presented in this paper, which provides the overview of work done for enhancing web search experience by creating similarity based page groups in the results. By doing so, we enable user to get similar information within a set/group of pages, instead of presenting user with scattered information. It is like clustering pages based on their similarities. Proposed method finds similarity among the results and group them according to similarity index. This method also utilizes semantically related concepts to user query, to enhance the results.

Categories and Subject Descriptors

H.3.3 [**Information Retrieval**]: Information Retrieval—*information filtering*; I.2.6 [**Articial Intelligence**]: knowledge acquisition—*clustering*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Online Page Categorization, Jaccard's Similarity Theory

Keywords

Web search, categorization, similarity, Jaccard's distance

1. INTRODUCTION

Web has turned into a huge repository on information, as consequence more information gathering is done using this medium. Thus, finding information has become more important and challenging task due to enormous size data that is on web and also due to its distributed nature. Web search engines and web directories generation has been a topic of interest of researchers for quite some time now. Thus, we have came a long way in key word based search engines. They can find thousands of pages (and documents) containing key words, from all over the internet, but as much useful it may sound, it isn't. The reason for that is any general query results in millions of pages, which makes the result list of pages seems like a haystack of information containing key words. Where user has to go through number of pages before realising he needs to update his query, to reach desired content. Now, even if user puts multiple key words in query, they are treated just as characters of string without any semantics or relation between them, Search engine still tries just to find these keywords within pages, which usually results in a clutter of unrelated pages around the useful

*Semester Project

†Data & Information Integration

content.
Where as, our system takes any genera query from user and results ins specific or categorized groups of similar or semantically related pages [2]. Thus, user can find desired content by going to desired content category without updating query to narrow down search results.
In order to create similar or semantically related clusters of pages, wikipedia, wordnet and bing were used at knowledge acquisition phase. crawling the web is very huge task as it needs processing of massive amounts of data. Thus efficient algorithms are needed for knowledge base building phase. Our system works in online knowledge acquisition, when any queries is seen for the first time by the system, and then acquired knowledge is stored in database, which is later used to help generate results when query is seen again as whole or as part of another query. Starting with single concept, our system gets related wikipedia page (if not available then related wordnet) and remove all formatting or natural language information which is not needed by system to acquire a concept vector from page. Nest, system performs a search for getting query results along with results for related concepts, gathered in previous step.These documents are also parsed to get word sets for each document. Now, system performs unsupervised clustering or get sets of disconnected graphs. system needs some distance measure to to calculate relation between pages, for this we made use of Jaccard's distance. After getting distances, we can cluster pages or just put pages in same category or group, if they have distance less than threshold to get similar pages. Result to user will be hieratically organised structure of these groups and categories.

2. RELATED WORK

Web page classification or categorization, is process of classifying pages into predefined categories, tough it can help in building web directories or in building Question-Answering (QA) systems but our tasks deals with unsupervised clustering of the search results to help improve searching experience of end user, because studies show that most of queries only contain 2 or 3 key words and boolean operators are nearly never used. As, web material is quite diverse even among specific key words, thus user cannot anticipate what is available in among set of results and how to form better and precise queries. Average web user has nor skill nor spare time to build complex boolean quires, which can narrow down search area. Ambiguity in natural languages in addition to choice of bad key words results in poor quality in search engine results. For example, Query "nearest bank" could be meant for river bank or financial institution . There are many techniques proposed to deal with this problem. Chekuri et al. [1] studied web page classification to improve results precision, This approach works only if someone can predefine categories for web and user is certain about category for his query.
As, search engine results are presented in plain list of pages but it can improve precision of results by presenting them in structure of organized categories or subcategories. An approach to classify pages into predefined categories and present user a organized hierarchical structure [3], Study showed category interface is liked by users batter than plain list results, and is more efficient in reaching user's desired information. Question answering systems use, Question classification and document classification to find correct set of documents to search for candidate answers. QA systems use predefined categories for classification. Thus, they can use classifiers for the task. [4]

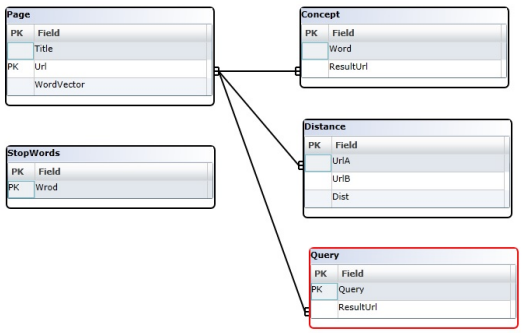

Figure 1: Database architecture.

3. MATERIALS AND METHODS

In this section we will first discuss proposed method and its implementation which is done using C# on Visual studio 2012 RC and using mySQL as Information store and also as graph db for storing page to page relations and similarity measure.

3.1 Motivation

Our system is motivated by, low precision in user queries and enhancing overall search experience. Thus, problem can be defined as, on the fly categorization or clustering of web pages based on conceptual similarity and semantic relations between web pages. Our system implementation fucus on providing platform which can categorise and reorganize plain list of search results in to a hierarchically categorized results.

3.2 Proposed Method

While doing survey of search engines, we came to know that most time consuming task in a search engine's life is crawling the web (on average google crawler, crawls about 2000 sometimes 20,000 pages in one day) and second most time consuming task is ranking results for a query. Many methods has been proposed for reducing time complexity using MapReduce framework and other modern technologies.
Initially, we decided to use Open Crawl data to avoid crawling the web and build complete search engine but it was out of focus of current problem, which was to deal with enhancing the experience using reorganization of results according to semantic similarity and relations.
Thus, we used bing search API to avoid implementations of key word search engine. For semantic analysis of words and concepts, we needed concept map of related concepts, for this we used related wikipedia page and parse for related concepts and wordnet to find similar word for better similarity measure. Than Using a database to store concept-page, and page-page relations along with there similarity/distance indexes. Based on similarity measure, we took similar pages and made groups for each set and present to user in a organized structure.

3.3 Algorithm

For purpose of search we are using Bing as back-end keyword search engine, using its JSON api and for related concept gathering we employed reverse search (using related wikipedia page to gather related concept) which gave us related concept and Sundice [5] (the semantic web index, it is web data integration platform, they gather semantic information form all over the web) for more of semantically related concepts (for example Pakistan is related to President of Pakistan and Embassador of Pakistan). Once, we have got the related concepts, now we can start performing search and grouping of pages.

Get initial concept list from wikipedia.

- Get wikipedia page, related to user query
- Parse page, and Remove
  - HTML tags
  - Stop words
  - Common words
  - Formatting information
  - Punctuation symbols
- Stemming or string normalization
- Result initial word vector

Perform search using wikipedia word vectors queries.

- Parse result pages, and Remove
  - HTML tags
  - Stop words
  - Common words
  - Formatting information
  - Punctuation symbols
- Stemming or string normalization
- Result word vector for each page

Calculate distance (Jaccard's distance) between pages.
Make page clusters based on distance.
Organize pages in a hierarchy.

For similarity or distance measurement between pages, we used Jaccard's distance [6], which is distance measure between pair of sets and is given by:
Jaccard's Index

$$J(A,B) = \frac{A \bigcup B}{A \bigcap B} \tag{1}$$

Jaccard's Distance

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \bigcup B| - |A \bigcap B|}{A \bigcap B} \tag{2}$$

Jaccard's index despite being easy to calculate, gives a good sense of similarity among pages. Groups of pages were made using getting disconnected set of graphs (node representing pages and link representing Jaccard's distance). These graphs (clusters) were made by removing any link with distance greater than threshold. So, we get similar pages in each sub-graph and can consider it as a category.

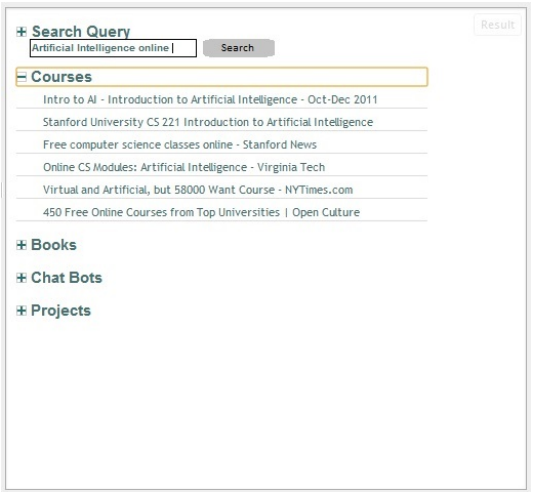

Figure 2: Result Page

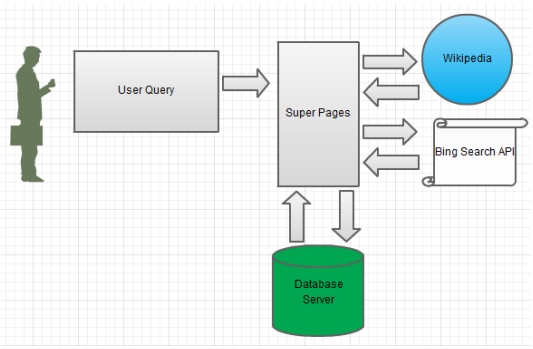

Figure 3: System Architecture

3.4 System & Database Architecture

System consists of a web server, C# back-end which communicates with with other services using web service interfaces and calculates results and web page similarities. Results along with word vectors are stored in Database.

The database used for system consists of five table. they are:

1. Stopwords: Containing stop words, common words and other words/marks which needs to be removed from text before forming word vector.
2. Page: Storing page tile, page url and related word set, for all pages yet visited.
3. Concept: Containing concept-url relations.
4. Distance: It contains url-url relation with distance measure between them.
5. Query: It contains query (can be multiple words) and resulting related concepts.

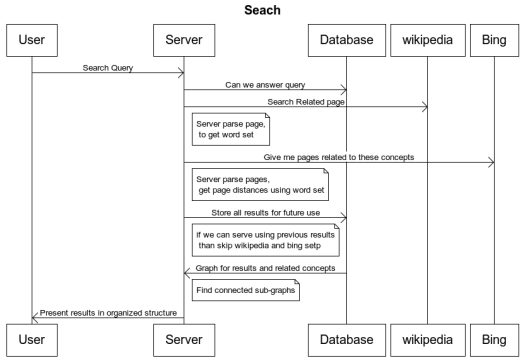

Figure 4: Sequence Diagram

Query	On query parsing	off-line web parsing
LUMS	45sec	1.25sec
Pakistan	60sec	2.30sec
University Trento	67sec	1.32sec
Samsung Galaxy mini	31sec	1.2sec

Table 1: Time comparison of different searches

4. RESULTS

In terms of time complexity, parsing results on query time is not most efficient. Thus we compare on query time similarity calculation to off-line or using already calculated similarity index. For example if parsing result pages for construction of word vector on average it takes from about $20 - 120 seconds$ and stated earlier most the search engines generates millions of search result. Parsing web pages is always slow, even big search engines, cache web and crawl over it all the time but that is not to get results for search query, it is to built search engine index, which is later used for fast query resolution.

Following table shows the time complexity of some queries. These results were taken on intel centrino laptop on a visual studio development web server.

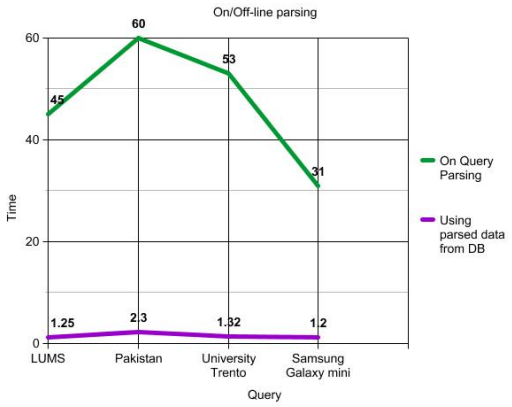

Figure 5: Results for offline vs on-query parsing.

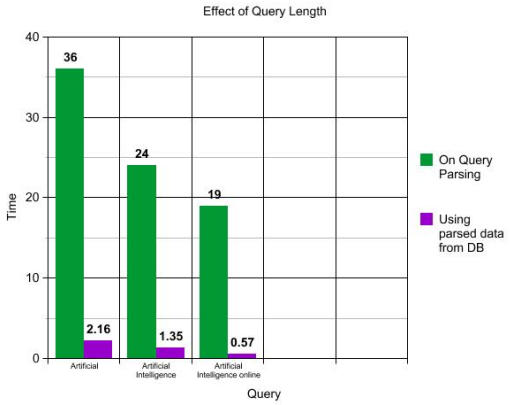

Figure 6: Comparison of different query lengths.

Experimentation shows that longer the query is or more specific user query is the less time it take, to get results.

5. CONCLUSIONS

Despite being slow when performing on query result parsing, methods does results in, reasonable performance while groups of similar pages (that is, if they exist in set of searched results). Currently the issue with employed method is, it lacks decision for where to draw a boundary and how to determine number of groups. Currently, every search result, is once treated as center for cluster, which come times replicates, already determined cluster, and some times results in formation of new cluster. For improvement and solution of these problems, complex clustering algorithms can be employed which can work on Jaccard distance. There arise and issue, that how similar pages should be to be considered similar not identical, as we can find much of similar information under different titles. we intend to replace search engine to a semantic search engine, which will help relating web pages more efficiently.

6. ACKNOWLEDGMENTS

I would like to thank project advisor (Yannis Velegrakis) for his supervision and guidance toward innovative solution.

7. REFERENCES

[1] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. Web search using automatic classification. In *In Sixth World Wide Web Conference*, 1997.

[2] W. chiu Wong and A. W. chee Fu. Incremental document clustering for web page classification, 2000.

[3] S. Dumais and H. Chen. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 256–263, New York, NY, USA, 2000. ACM.

[4] U. Hermjakob. Parsing and question classification for question answering. In *Proceedings of the workshop on Open-domain question answering - Volume 12*, ODQA '01, pages 1–6, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[5] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *Int. J. Metadata Semant. Ontologies*, 3(1):37–52, 2008.

[6] Wikipedia. Jaccard index, 2004. [Online].