Indian Journal of Science and Technology, *Vol 8(7), 635–641, April 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Novel Architecture of Perception Oriented Web Search Engine based on Decision Theory

Vinit Kumar[1], Niraj Singhal[2]*, Ashutosh Dixit[3] and A. K. Sharma[4]

[1,2]Shobhit University, Meerut, India; sonia_niraj@yahoo.com
[3]YMCA University of Science and Technology, Faridabad, India
[4]BSAITM, Faridabad, India

Abstract

The number of active web pages increases exponentially. According to the survey, the web has 14.3 trillion active web pages. The problem faced by present search engines is difficulty in returning relevant information. The current search engines do not perform semantic search and are not capable to return results based on user's perception. In this paper a perception based search engine is proposed that returns results as per the user point of view. To achieve semantic searching, a knowledge base is constructed which stores knowledge in the form of predicates. To extract knowledge from knowledge base, decision theory is used that does not restrict to any specific domain.

Keywords: Decision Theory, Knowledge Base, Mental Vision, Perception, Semantic

1. Introduction

The Web is a huge repository of interlinked hypertext documents accessed via Internet. Through a web browser a user can view web pages that contain text, videos, images and other multimedia information and navigate between them via hyperlinks. To retrieve information from the web, the activity begins either by typing the Uniform Resource Locator (URL) of the page, a list of keywords into a web browser or by following a hyperlink to that page or resource. The web browser then begins a series of communication messages, in order to retrieve and display the relevant information. Due to the enormous size of information resource on the web, it has become more difficult to retrieve relevant information. A web search engine helps users to find information so it is necessary for a search engine to locate the relevant information that is needed by a user. Web search engines are used by a variety of users who fetch their queries in several different formats. For example, the users willing to search for first president of USA fetch their queries in many forms like 'First President of USA', 'USA First President' and 'USA President First'. Since a general web search engine performs only keyword matching, the results returned by a search engine for various forms of same query are different. Moreover, it is also found that a search engine returns millions of results of which only few are relevant to the user query[2]. Since, the current search engines do not perform semantic search, they are not capable to return results based on user's perception. This paper presents a decision theory based search technique that returns results as per the mental vision of the user i.e., the information he is intended to search.

2. Related Work

A search engine contacts web server to retrieve all web pages and updates its repository (a stored collection of web documents) to fulfill the need of users. To retrieve the information from the web, information seekers begin their activity by typing a query, typically a list of keywords to search engine interface. This entire task is done at search engine side.

Author for correspondence

There are many techniques used to match the web pages for a user query. One is Boolean retrieval model in which users can pose any query which is in the form of a Boolean expression of terms, i.e., in which terms are combined with the operators AND, OR and NOT. Search engine based on Boolean retrieval views each document as a set of words. For example to search (lion or tiger) and India, the search will yield document containing "lion" and "India", "tiger" and "India" or all three terms. It will not yield documents containing only "lion", only "tiger" or only "India".

Since a search engine has to parse all documents available on the web, it requires large computational power and time, and results in a very large number of keywords. To resolve this problem, search engines create index to store keywords. Many search engines use inverted index to evaluate a search query and to locate the documents quickly, and then on the basis of rank; judge the relevance of the documents to provide the results[2]. To find relevance, a search engine collects relevance assessments. Due to the involvement of human beings, this is a time-consuming and expensive process. A standard approach for relevance assessment is pooling[2] where relevance is evaluated over a subset of the collection to return top 'k' documents. As it involves human beings, a common measure for agreement between judges is done by kappa statistic[2]. The rate of chance agreement is computed as follows,

$$\mathbf{kapp} = \frac{\mathbf{P(A) - P(E)}}{\mathbf{1 - P(E)}} \quad \text{Eqn. (1)}$$

Where P(A) is the proportion of times the judges agreed and P(E) is the proportion of times they would be expected to agree by chance.

Problem lies with the relevance based assessment is, to make relevance in case of duplicate documents found on web. Marginal relevance requires returning documents that exhibit diversity and novelty[2]. Having chosen or ranked the documents, to present a results list that will be informative to the user. The standard way of doing this is snippet[1,7] a short summary of the document which is designed so as to allow the user to decide its relevance. Two basic kinds of summaries are static summary and dynamic summary. Static summary remains same as the query, and dynamic summary is modified according to the user's information need. Dynamic summary improves the ability to find more relevant result but cannot be precomputed[1]. Query refinement plays a vital role to find relevant results. There are two methods used for query refinement to extract better relevance feedback i.e., Local methods and Global methods. Local methods are used to alter a query relative to the documents that initially be published to match the query. Some basic local methods are Relevance feedback, Pseudo relevance feedback and Indirect relevance feedback. In Relevance feedback user provides feedback to initial result sets produced by the system. Then system re-computes itself to determine which pages are relevant or not based on user feedback. The Rocchio algorithm1 for relevance feedback is used for this purpose. As per Rocchio the modified query is,

$$\bar{q}_{m} = \alpha \bar{q}_0 + \beta \frac{1}{|D_r|} + \sum_{\vec{d}_j \in D_r} \vec{d}_j - \beta \frac{1}{|D_{nr}|} + \sum_{\vec{d}_j \in Dn_r} \vec{d}_j$$

Eqn. (2).

Where is the original query vector, Dr are the set of relevant pages, Dnr are the set of non relevant documents, and α, β, and γ are weights attached to each term, used to make balance between judged documents and query.

Relevance feedback plays important role to increase recall in information retrieval. An example of Rocchio algorithm is shown in Figure 1, in which documents have been labeled as relevant or non-relevant. The initial query vector is pointed to this feedback where 'x' represents non relevant documents and 'o' represents relevant documents. Naive Bayes probabilistic model is also used to coordinate the task of relevance feedback. This is done by determining the probability of a term appearing in a document based on whether it is relevant or not[8].

Pseudo relevance feedback method first provides most relevant initial set of documents based on rank then uses the phenomena of relevance feedback[6,9]. Indirect relevance feedback uses indirect sources to determine the relevancy of documents. For example relevancy of a document in indirect method determines based on the number of hits on a link[1,10] generally used for highly volume data systems like for web search engine. Global methods for query refinement1 are vocabulary tools for query reformulation and query expansion[16]. In vocabulary tools for query reformulation method, information retrieval system suggests some terms from thesaurus (a collection of vocabulary). In query expansion, search engine suggests some related queries in response to a query[1,11,12]. This task is done using some form of thesaurus. For each term 't' in a query, the query can be automatically expanded with synonyms and related words of 't' from the thesaurus.

All the methods discussed above depend on user involvement such that to give relevance feedback, creation of thesaurus and use of controlled vocabulary maintained by a human being. None of these methods are able to provide most relevant results considering the sense behind words entered by users.

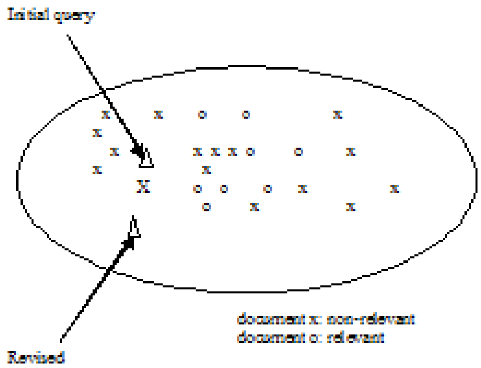

Figure 1 An example of Rocchio algorithm.

3. Proposed Work

A search engine returns millions of the results in response a user query. To show most relevant pages between the different possible outcomes for a search engine it is necessary to refine the query in a manner that user should get whatever he is expecting in outcome. So, a utility theory with probability theory plays an important role to perform this task. As per Utility theory, every state has a degree of usefulness or some utility. So using probability theory, selects useful states. This combined approach is also called decision theory[13]. Decision theory is used to select highest expected utility, average over all the possible outcomes. In proposed architecture a decision theory agent selects rational outcomes. The proposed work is carried out to provide only relevant results to a user query. To extract knowledge from knowledge base various techniques are available, like resolution inference rules, forward chaining, backward chaining and unification etc. The problem with these methods is that they suit only for a specific domain (like medical diagnosis system). In this proposed architecture decision theory used i.e., not restrict to any specific domain. It is based approach for relevance judgment and returns same result set for same query fetched in different forms. The proposed architecture of perception based web search engine based on decision theory is shown in figure 2. Several components of proposed architecture (see Figure 2) are Crawling module, Indexing module, Queries parser module, Ranking module, Knowledge representation module, Perception based query refinement module and Result matcher module. Functioning of all modules is as follows:

3.1 Crawling Module

It downloads web pages from the web, read contents of web pages, follow all valid links at crawled web pages, periodically return to sites to check the information that has changed and stores all useful web pages and information to the search engine repository.

3.2 Indexing Module

Extracts all the uncommon words from the web pages downloaded by the crawler and record the URL where each word has occurred. The result is stored in a large table containing URLs pointing to pages in the repository where a given word occurs.

3.3 Query Parser Module

Translates the user specified keywords into a query that is submitted to the perception based query refinement module.

3.4 Ranking Module

Since the user query results in a large number of web pages, now it is the responsibility of ranking module to decide which web pages to be displayed at top in the result set.

3.5 Knowledge Representation Module

Full architecture of Knowledge base representation module is shown in Figure 3. It represents knowledge in the knowledge base in term of facts and rules. Knowledge representation is a tool that accurately uses a set of symbols to represent a set of facts within knowledge base. Knowledge can be represented using semantic net, frames or decision network. Because XML representation is considered best to extract semantic relationship15,17,18,19 so there is a need to parse HTML documents into XML format. There are two functional modules (see Figure 3) that complete the above task. First, XML converter takes HTML documents as input and produces their equivalent XML. Second, predicate generator, form semantic net from XML and stores this information in the form of predicates in the knowledge base.

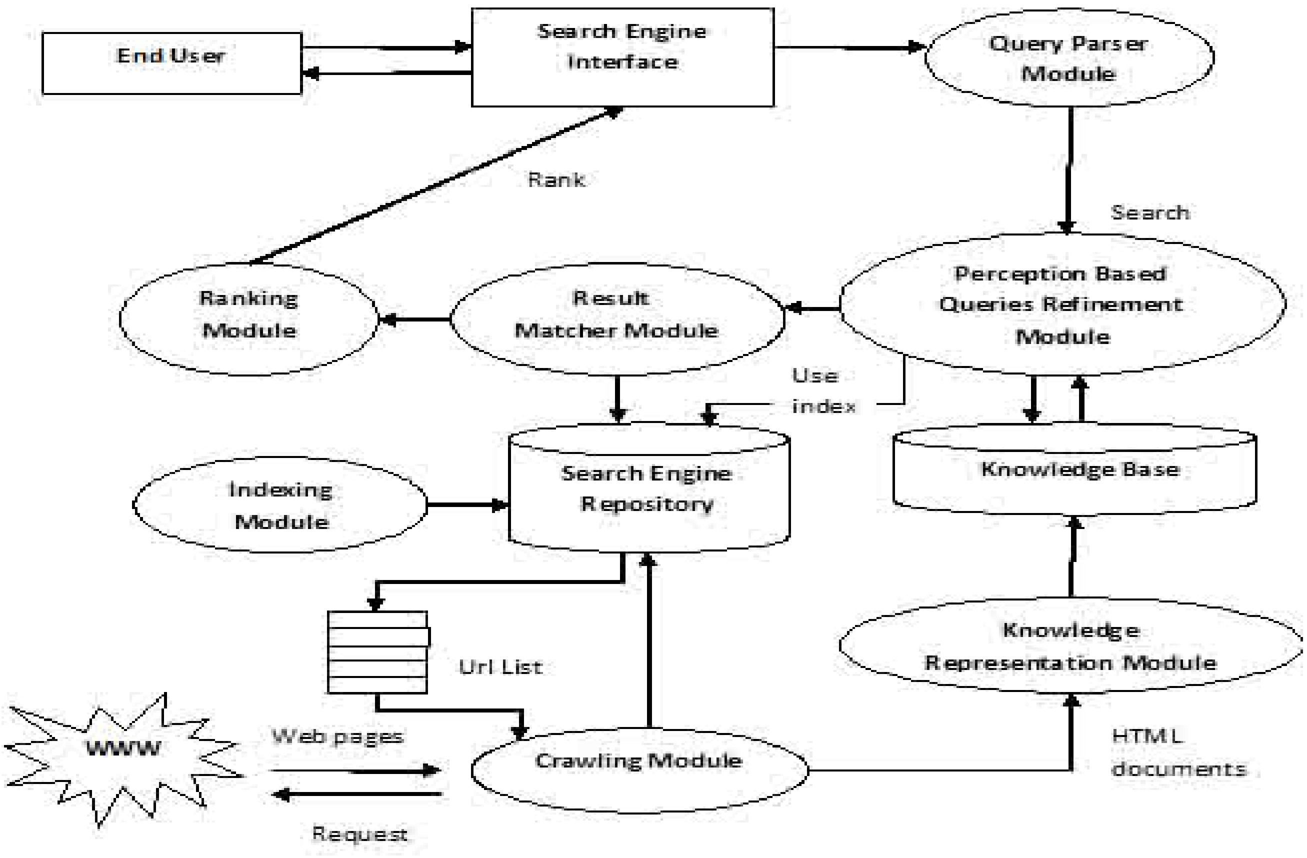

Figure 2 Architecture of perception oriented web search engine.

3.6 Perception Based Query Refinement Module

Uses index (stored on search engine repository) to retrieve all relevant page_ids related to query. Then fetches predicates from knowledge base corresponding page id retrieved from index. Apply decision theory to predicates and return page_ids having highest utility and better probability to meet specified requirement.

3.7 Result Matcher Module

Retrieve appropriate web pages based on page_id returned from perception based query refinement module. The retrieved web pages are passed to ranking module. Databases used are Search engine repository and Knowledge base (Figure 2).

3.8 Search Engine Repository

Stores useful web pages along with page ids crawled by search engine, index table and other metadata.

3.9 Knowledge Base

A collection of sentences that store knowledge in the form of rules and facts that have some semantic meaning related to any specific domain.

Knowledge Representation in form of predicates is shown as follows:
First President (George Washington, United State)
Ended (Revolutionary War, 1789)
First President (George Washington, America)
Considered (Jone Hanson, First President)
Patriot (Jone Hanson, America)
Son of (Charles County, Hanson)
Carrier (Hanson, Political)
Lived (Hanson, Marry Land)

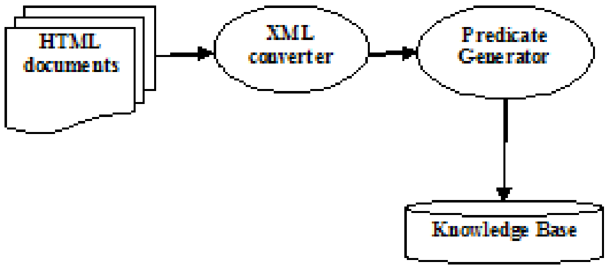

Figure 3. Components of knowledge representation module.

To evaluate a query procedure begins with extracting rules and facts in form of predicates corresponding to page_id retrieved from search engine repository. Now decision theory use utility function and probability theory to chose page_id with highest utility and with better probability preference. The page_ids chosen by decision theory are passed to result matcher module that retrieve all web pages corresponding to these page_ids. Now result matcher module handover all web pages to ranking module. Ranking module computes rank, to determine order in which web pages will be displayed in the result set.

Algorithm to retrieve most relevant pages based on user perception is as follows:

Extract_relevant_pages

1. Input query to search engine interface.
2. Query parser on receiving query parse this query in more optimized form and passes it to perception based query refinement module.
   a. Query refinement is done by applying decision theory algorithm that returns page_ids based on highest utility and probability preference, and then all page_ids are returned to result matcher module.
   b. Now, result matcher retrieves all web pages based on page_ids returned from perception based query refinement module and pass to ranking module.
   c. Now rank module determines the rank of web pages.
3. Return web pages and shown as result set.

Algorithm for decision theory evaluation is as follows:

Decision_theory_evaluation {

1. Set the evidence keywords for the current state.
2. For each page_id chosen by perception based query refinement module
   a. For each predicate (rules) calculate utility U_{pr} done by matching each current state with predicate terms.
      If (predicate_term=current state)
      $U_{pr} = U_{pr} + 0.1$
      Else
      $U_{pr} = U_{pr} + 0.0$
   b. Calculate utility ( ) value and probability for a page.

$$U_{Pi} = \sum U_{pr} \quad \text{Eqn. (3)}$$

$$PR_{Pi} = Max(U_{pr}) * \frac{10}{qt} \quad \text{Eqn. (4)}$$

Where, qt is number of query terms without stop words.

3. Now determine excepted highest utility of web pages using following formula.

$$\mathbf{EHU(P)} = \sum_{i=0}^{n} \mathbf{Result}\frac{U_{Pi}}{\square}\mathbf{n}$$

Where, n is total number of documents.

4. Now, select page_id with respect to highest utility EHU(P) and probability and return to result matcher.

URLs of various documents retrieved for a query ‘first president USA’.

On applying decision theory, the documents that have highest utility (EHU) and probability >0.66 with consideration ‘yes’, fulfil the requirements of semantic searching. One more benefit obtained by this proposed work is, every time a user gets same set of results for a query fired in different forms. The number of results found by present search engine for a ‘first president USA’ query in fire different form like ‘USA first president’, ‘USA president first’, ‘who was first president USA’ are 766000000, 1060000000,79200000 and 1070000000 consecutively. It returns same eight results out of fourteen (considered for experimental overview) for the above query fired in different forms.

4. Conclusion

The perception based search engine based on decision theory outlined in this paper lays the foundation of semantic web search engine. The proposed methodology based on decision theory provides most relevant results for a user query. Benefit obtained by this work is that user gets same result set for a query fire in different form. . This technique also reduces the network bandwidth requirement to carry user data by retrieving only relevant web pages to

Table 1. URLs of Pages retrieved with page_ids

URLs	Page_id
http://voices.yahoo.com/first-american-president-george-washington-john-849195.html	1
http://www.constitution.org/hist/first8pres.htm	2
www.marshallhall.org/hanson.html	3
http://www.history.com/this-day-in-history/john-hanson-so-called-first-president-dies	4
http://en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States	5
http://www.whitehouse.gov/administration/president-obama	6
http://en.wikipedia.org/wiki/President_of_India	7
http://en.wikipedia.org/wiki/President_of_the_United_States	8
http://www.freerepublic.com/focus/bloggers/1771850/posts	9
http://www.whitehouse.gov/about/presidents	10
http://www.history.com/this-day-in-history/first-us-president-elected	11
http://www.straightdope.com/columns/read/1551/was-george-washington-not-the-first-u-s-president	12
http://www.biography.com/tv/classroom/us-presidents-in-order	13
http://www.enchantedlearning.com/history/us/pres/list.shtml	14

Table 2. Documents with utility value and probability

Page_Id	Utility	Probability	EHU	Consideration
1	1.0	3/3=1	0.65	Yes
2	1.3	3/3=1		Yes
3	0.6	3/3=1		Yes
4	0.4	2/3=0.66		No
5	0.6	2/3=0.66		No
6	0.3	2/3=0.66		No
7	0.7	2/3=0.66		Yes
8	0.5	2/3=0.66		No
9	0.9	3/3=1		Yes
10	0.4	2/3=0.66		No
11	0.6	3/3=1		Yes
12	0.8	3/3=1		Yes
13	0.7	2/3=0.66		Yes
14	4.5	1/3=0.33		No

full fill the user query requirement. XML representation is best suitable for knowledge representation but web is found unstructured in nature so it is quite difficult to parse these documents to XML representation need further researches to carry this task.

5. References

1. Manning CD, Raghavan P, Schutze H. Information Retrieval System, 1st edition. England: Cambridge University Press; 2009.
2. Singhal N, Dixit A, Agarwal RP. (2012). User perception based inverted index for web search engines. International Journal of Contemporary Research in Engineering and Technology. 2012; 2(1):39–44.
3. Dragomir Q, Vahed RR. Novel methods in information retrieval. Ann Arbor: Department of EECS, University of Michigan; 2008. p. 1–63. Technical Report CSE-TR-546-08
4. Carbonell JG, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Informational Retrieval. New York; 1998. p. 335–6.
5. Buckley C. Implementation of smart information retrieval system. Cornell University; 1985 May. p. 1–40. Technical Report TR-85-686
6. Carbonell JG, Geng Y, Goldstein J. Automated query relevant summarization and diversity-based reranking. Proceedings of 15th International Joint Conference on Artificial Intelligence, Workshop: AI in Digital Libraries. Nagoya, Japan; 1997. p. 9–14.
7. Overwijk. enerating snippets for undirected information search. Proceedings of 9th Twente Student Conference on IT. Enschede; 2008.
8. Overwijk A, Nguyen D, Hauff C, Trieschnigg RB, Hiemstra D, de Jong FMG. On the evaluation of snippet selection for information retrieval. Proceedings of the 9th Cross-language Evaluation forum Conference on Evaluating Systems for Multilingual and Multimodal Information Access; 2007. p. 794–7.
9. Liu Y, Yan H. (2013). A new probabilistic model for bayes document classification. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering. Hangzhou, P. R. China; 2013. p. 1239–42.
10. Rocchio J. Relevance feedback in information retrieval. The SMART Retrieval System: Experiments in Automatic Document Processing; 1971. p. 313–23.
11. Cao G, Nie J, Gao J, Robertson S. (2008). Selecting good expansion terms for pseudo-relevance feedback. Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2008. p. 243–50. 12. Metzler D, Croft BL. Concept expansion using Markov random fields. The Proceedings of SIGIR; 2007. p. 311–8.
13. Ogilvie P, Callan J. The effectiveness of query expansion for distributed information retrieval. School of Computer Science Carnegie Mellon University Pittsburgh, PA; 1999. p. 238–45.
14. Russell JS, Norvig P. Artificial Intelligence a Modern Approach, 2nd edition. Pearson Education; 2008.
15. Tyagi N, Rishi R, Agarwal RP. Semantic structure representation of html document suitable for semantic document retrieval. Proceedings of International Journal of Computer Applications. 2012; 46(13): 39–43. ISSN: 0975-8887.
16. Taneja N, Aggarwal K, Aggarwal N. Ontology based conjunctive query expansion. Proceedings of International Journal of Emerging Trends in Engineering and Development. 2012; 6(2):81–8.
17. Sekine S. The domain dependence of parsing. Proceedings of the fifth Conference on Applied Natural Language processing. Stroudsburg, PA, USA; 1997. p. 96–102.
18. Noah SA, Che AA, Zakaria LQ. A semantic retrieval of web documents using domain ontology. Proceedings of International Journal of Web and Grid Services. 2005; 1(2):151–64.
19. Sekine Proteus Project - Apple Pie Parser. Available from: http://nlp.cs.nyu.edu/app (Corpus based Parser); 2006.