Fuzzy Web Information Retrieval System

Joseph Lee
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202
theus81@gmail.com

Eunjin Kim
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202
ejkim@cs.und.edu

Abstract

In this paper, a fuzzy web information retrieval system is developed. The system uses many of the tools and methods involved in fuzzy logic and fuzzy set theory, along with standard algorithms involving information retrieval on the web. The results of the ranking algorithm use the fuzzy relational BK-products, fuzzy thesauri, and fuzzy closure properties for purposes of retrieving relevant documents to a user query.

Along with the fuzzy inference engine, the other subsystems of a search engine are also developed. A crawler is developed that adheres to standard internet etiquette rules. An inverted index database is used to store the internet document information retrieved by the crawler and an interface is also provided to allow users to search for gathered information located within the database.

The results of the developed system are compared with existing search engines. The benefits of using a fuzzy information retrieval system are discussed, along with the issues that arise from using these techniques.

1 Introduction

In this paper a fuzzy web information retrieval system was developed. A fuzzy web information retrieval system is an information retrieval system that utilizes the tools involved in fuzzy set theory and fuzzy relational theory. Fuzzy logic and fuzzy set theory was first proposed in 1965 by Lotfi Zadeh as an alternative to standard Boolean logic [1]. Fuzzy logic deals with the existence of imprecise or vague data that may be neither entirely true nor false, but somewhere in between. Fuzzy logic extends and becomes an alternative to Boolean logic.

The design of the fuzzy information retrieval system is discussed along with its various subsystems that comprise the methods used to retrieve documents from the web. Major issues with information retrieval are mentioned along with the results of the design of the fuzzy web information retrieval system.

1.1 Objectives

Document retrieval based on relevance to a user query is still an ongoing field of research. In response to the issues that relevance still poses to information retrieval, this research attempts to address these issues with the development of a fuzzy information retrieval system. This system contains all the functionality of a standard search engine, including a web crawler and indexer. The fuzzy information retrieval system contains a ranking operation which invokes methods derived from the fuzzy relational BK Products developed by Bandler and Kohout [2-4] to rank documents retrieved by the information retrieval system. The documents contained within this system will be extracted from the internet. In this way, the system will work similar to a search engine.

The goal of this study is to rank documents retrieved from the internet that are more relevant to a user query with a higher value than those documents that are less relevant to a user query. This research uses new tools to rank documents within a repository to retrieve better search results. Using the tools in fuzzy information retrieval will aid in the retrieval and ranking of relevant documents from the web.

2 Information Retrieval on the Web

Retrieving information from the web can prove to be difficult because of the size and abstractness of data contained on the web. Approximations for 2011 estimated the web to be as large as 50 billion web pages or more [5]. Web retrieval is made increasingly difficult when adding in factors such as word ambiguity (where a single word can take on multiple meanings), and the large amount of typographical errors contained within web information. It is estimated that one in every two-hundred words, on an average web site, will contain a textual error [6].

2.1 Issues with Information Retrieval

There are several key issues involving information retrieval. These issues are relevance, evaluation, and information needs. However, these are not the only issues involving information retrieval. Other issues such as performance, scalability and occurrences of paging update are other common information retrieval issues [7].

Relevance is the relational value of a given user query to the documents within the database. Relevance of a document is normally based on a document ranking algorithm. These algorithms define how relevant a document is to a user query by using functions that define relations between the query given and the documents collected in the index.

The evaluation of the feedback given by the information retrieval system is another issue with information retrieval. The behavior of the system may not meet the expectations of the user or the documents returned from the system may not all be relevant to a query. Depending on the system and the user, the results of a query should be in a format that most fits the data being searched and returned.

Information needs is how the user interacts with the information retrieval system. The data within the system should be able to be accessed easily and in a way that is convenient to the user. Retrieving too much information might be inconvenient in certain systems, also in other systems not returning all relevant information may be unacceptable.

2.2 Document and Query Management

As noted, managing volumes of information from the web can be difficult due to the volume of documents the web contains. This leads to an even larger problem when trying to retrieve the most relevant results to a query. A simple retrieval query can return thousands of documents, many of which are loosely related to the original retrieval criteria. Natural language and word ambiguity can also add distress to document retrieval. To overcome this, an information retrieval system needs to have good query management along with the ability to give weight to documents that are more relevant to the user's query, and present those results first.

Development of an efficient document ranking algorithm is crucial to an information retrieval system. There are many methods used to rank documents to a given user query. Common document ranking algorithms are term counting, user relevance feedback, popularity of the document, size of the document, term location within the document, and how often a document is linked [8].

2.3 Fuzzy Information Retrieval

Fuzzy information retrieval systems utilize the tools defined in fuzzy logic and fuzzy relations to infer the best results to a user query. Unlike Boolean systems, fuzzy systems are most effective when dealing with data that may display a degree of membership. In

fuzzy systems, objects described in terms of their properties which characterize the objects are assigned relational membership values to show relevancy from properties to objects or vice versa. These values are different than using probability. In probabilistic systems of the discrete case, count, i.e. the total number of an event, is used to compute the probability that a relationship exists, whereas in fuzzy systems, a membership value is used to determine a weighted relational mapping [9].

The scope of information retrieval was greatly expanded with the development of the BK-Products of fuzzy relational compositions established by Bandler and Kohout [2]. These relational methods allow for semantically based deduction from input to results through fuzzy logic of implication. This means that search queries can be translated through membership values to have linguistic meanings.

3 Fuzzy Information Retrieval System Structure

The fuzzy information retrieval system in this paper is divided into four individual components. These components are the crawler, the indexer, the fuzzy ranking settings, and the user's search component.

3.1 Web Crawler

The most common way of gathering web documents is to use an intelligent agent to gather the documents. These agents are most commonly referred to as crawlers, but are also known by other names such as spiders, ants, automatic indexers, bots, web robots, and worms [5].

Crawlers are given an initial starting location on the web called the seed URL. Crawlers then proceed to navigate through the web extracting URLs that occur on the document they are currently visiting. Developing a crawler is a difficult task, as there are many problems a designer might not foresee that could cause the crawler to crash or cause unstable and unpredictable behavior [6].

Crawlers must be sure to avoid overloading web servers with their processing and page accessing. They must provide as little overhead as possible as they proceed through the web. In response to etiquette issues and to prevent the crawler from potentially being banned from a specific server, the crawler in this paper has a built in user set delay. This artificial delay prevents the crawler from traversing websites too quickly which could be seen as abusive to server bandwidth. Additionally, many sites follow the *robots.txt* standard. This is a practice that notifies crawlers of documents that are safe to retrieve within a domain, and others that may not be worth gathering. This standard also protects sites from being overloaded or generally abused by crawlers [10].

The crawler developed within this paper also allows for a maximum link retrieval count. This enables the user to stop the crawler after it has retrieved up to a specific amount of web documents from a particular seed URL. The crawler will also adhere to standard web etiquette defined by the *robots.txt* standard.

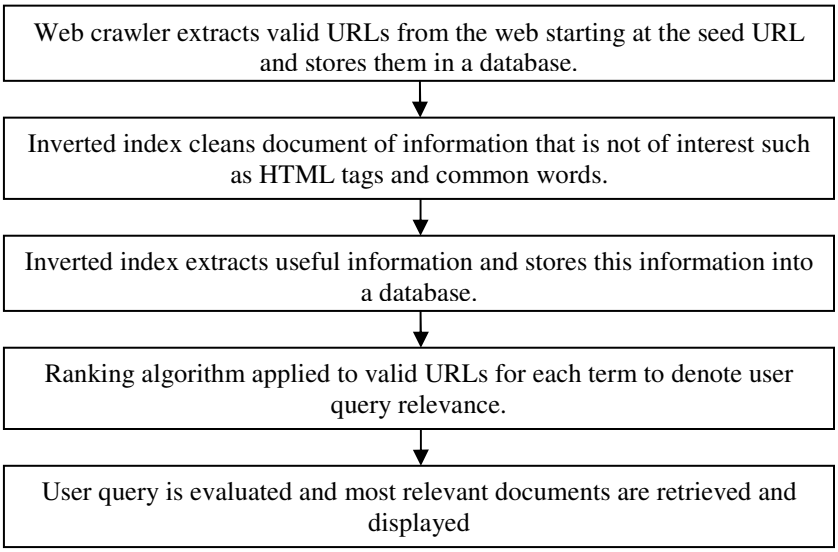

Figure 1: Search Structure

3.2 Web Document Indexing

After pages are retrieved by the crawler, they are passed into a page repository in the form of a database for further analysis by the indexer. Indexing that is done automatically is known as automatic indexing, and is the primary method used to index web documents. Indexing involves sorting the gathered documents by content. The indexer determines the subjects of each document to allow for retrieval by topic.

An *inverted index* is a mapping of documents to terms, rather than terms to documents. When searching for information of interest, a user is trying to retrieve documents by entering the terms into the information retrieval system that they have interest. By inverting the index, and mapping documents to terms, the system can select the terms the user has interest and retrieve the corresponding documents involved with those terms. An inverted index can be viewed as the locations where the terms were found.

To begin this process one must determine what content each document contains. To accomplish this, the indexer must extract the keywords and phrases from each document. The indexer must also remove unimportant words from the documents. Examples of insignificant words would be articles, prepositions, conjunctions and pronouns. In [6] the author has shown that up to 40-50% of the content within the document can be regarded as insignificant and can be skipped over when indexing. After the insignificant words are removed, the remaining terms can be collected and analyzed then they are able to be placed within the index accordingly. In this system, the use of regular expressions has been employed to help extract the information from the documentation coding and remove information that is not important to the user. The steps involved in the indexing of documents and terms within this system are to retrieve a URL from the list of valid URLs from the database, extraction of document information, removal of insignificant words, insertion of document in relation to specific terms into the inverted index, and update inverted index with new weights on documents.

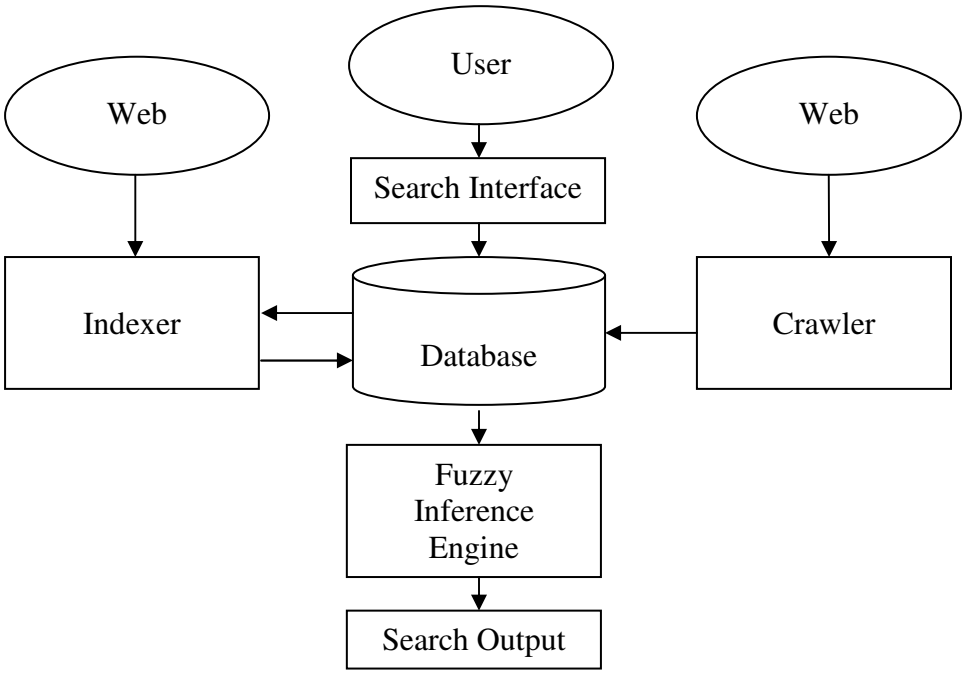

Figure 2: System Flow Diagram

3.3 Ranking using the Fuzzy Relational Inference Engine

This portion of the system is dedicated primarily to the flow of query evaluation and document ranking. This subsystem has two primary functions—the generation of the fuzzy relational matrix of terms to documents, and the generation of the fuzzy thesaurus which gives relationship from search terms to other terms, and the ranking of documents. These matrices are the primary tools used to rank documents in relation to a user query. The steps involved in defining the ranking of the fuzzy information retrieval system are as follows:

- The matrix defining the associations between terms and documents is generated from the information gathered by the indexer that is stored in the database.
- The thesaurus that defined the relationships from terms to terms is generated based on information gathered from the database. The user chooses the implication type used in generating this matrix, along with the fuzzy criteria used to evaluate the results of the fuzzy product. A transitive function can be chosen to compute the transitive closure of the search.
- The fuzzy implication, criteria, and product are chosen to define how the association between the two matrices is computed.
- The results are flattened into a single membership relation from the user query to the documents within the database.
- An α–Cut is applied to the list of retrieved documents to retrieve only relevant documents. An α-cut is a chosen value where any membership value of a document that is less than the α-cut is unrelated, and any membership value equal to or greater than that the α-cut is considered relevant to the user query. The remaining documents are ordered by rank and displayed to the user.

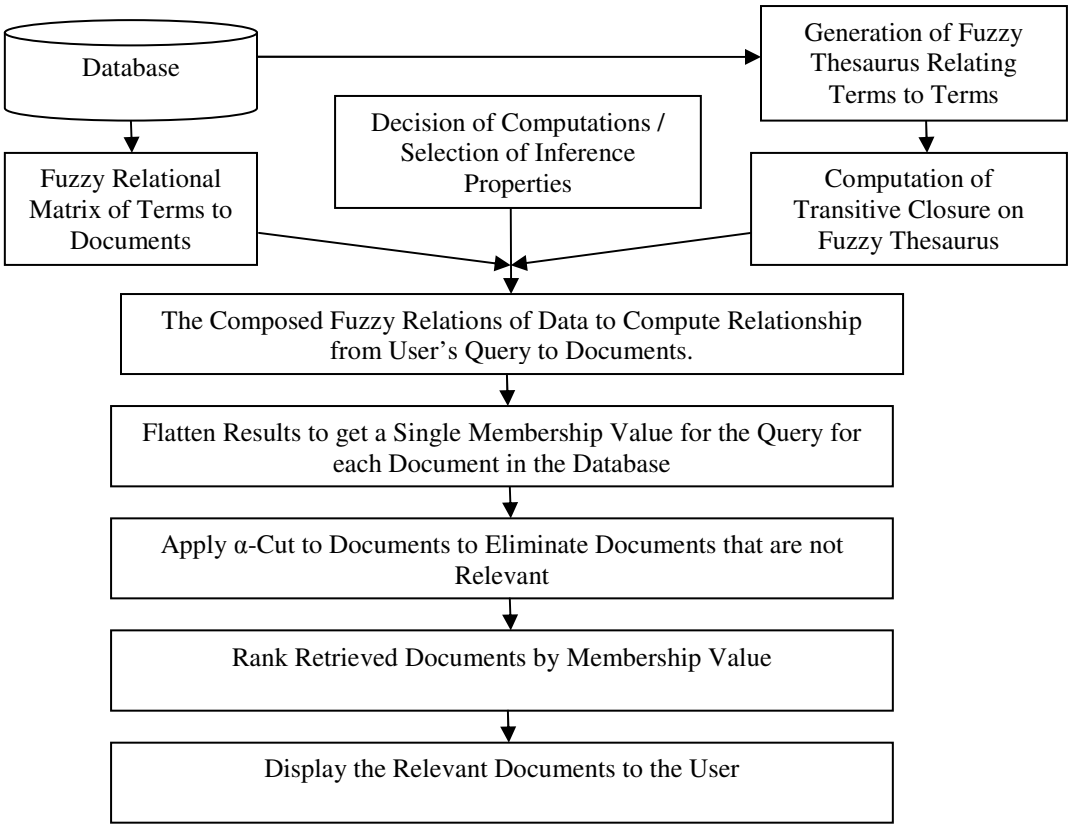

Figure 3: Procedure of the Fuzzy Inference Engine

A common technique used in giving weights to relationships between documents and terms is known as tf-idf or *term frequency–inverse document frequency*. This method of term weighting takes factors about the document repository and incorporates them into the weighting algorithm. One function using the logic from tf-idf is known as the Okapi BM25 ranking function developed by Robertson [11, 12]. In this paper, the variant of idf used for determining the weight of a term t_i is computed using the equation

$$\mathrm{idf}(t_i) = \log \frac{\mathrm{N} - \mathrm{n}(t_i) + 0.5}{\mathrm{n}(t_i) + 0.5}$$

where N equals the total number of documents in the database and $\mathrm{n}(t_i)$ equals the total number of documents that contain t_i in the database. This version of idf is then used within the Okapi BM25 ranking algorithm

$$\mathrm{BM25}(t_i) = \mathrm{idf}(t_i) * \frac{\mathrm{f}(t_i, d) * (k+1)}{\mathrm{f}(t_i, d) + k * (1 - b + b * \frac{|d|}{a})}$$

where $\mathrm{f}(t_i, d)$ is the frequency of a term t_i in a document d, $|d|$ is the length of document d, and a is the average length of all the documents in the database. The variable k is a variable that controls how important term frequency is. The larger value k is the more

important term frequency is when evaluating a relation. The variable b is the normalization of document length to term frequency, $0 \leq b \leq 1$. $b = 0$ will not normalize the term frequency with the length of the document at all. while $b = 1$ will inversely scale the term frequency with the length of the document. A value of $k = 2$ is used in this study and a value of $b = 0.75$ is used.

The values generated using Okapi BM25 are not strictly between the values 0 and 1, therefore a normalization function needs to be applied to the results. The maximum value for t_i given any document will scale all values appropriately to use as membership values.

$$x(t_i) = \frac{\mathrm{BM25}(t_i)}{\mathrm{BM25}(t_{max})}$$

where $\mathrm{BM25}(t_{max})$ is the maximum value generated for any document. Using these results, a relational matrix of terms to documents in fuzzy normal form can be populated for evaluation by the fuzzy inference system. These results are stored within the database yielding up to $t \times d$ total relations where t is the total amount of unique terms known, and d is the total amount of documents within the system.

3.4 User Search Structure

The user search structure is mainly the interface, layout, and format of the results of a user query. When a system returns the results of a query, it should also be in the format the user was expecting. Different systems may have different expectations for the presentation of their results. A database system may be required to return every relevant document to a user query. In this case the system knows that its data is complete so that all information can be analyzed. However, expecting a web search engine to return every document relevant to any query could be inconvenient to the user.

The information retrieval system developed in this paper has a very basic structure. The results displayed will be displayed in order from most relevant to least relevant until the user entered α-cut value for a document has been reached.

4 Evaluation

There are many factors that must be considered before the construction of a fuzzy web information retrieval system could be attempted. Since fuzzy inference is so engrained within matrix computations, many functions find themselves having a high complexity such as the worst case spatial computation of $O(n^2)$ where n equals the total number of unique words extracted from documents and stored within the system. For this reason, any word that only appeared once within a document was removed when evaluating relevant document information. Doing this dramatically reduced the amount of computations necessary for fuzzy inference and information retrieval. Likewise, the time complexity in the worst case for computing the major matrices was also $O(n^2)$. These matrices include the basic thesaurus, and the fuzzy matrix containing terms to documents relations. The most computationally expensive portion of the system was the transitive

function. Computing transitive closure of a thesaurus had a computational worst-case of $O(n^3)$. This amount of information makes processing very expensive both spatially and temporally. In [13] the author also notes similar expensive computational limitations. The spatial considerations for the thesaurus are displayed in Table 1.

Terms	Spatial Requirements
500 distinct terms	Up to 250,000 relations
2,500 distinct terms	Up to 6,250,000 relations
12,500 distinct terms	Up to 156,250,000 relations
62,500 distinct terms	Up to 3,906,250,000 relations

Table 1: Thesaurus Matrix Spatial Considerations

The results in this experiment were similar to other ranking algorithms; this is partially due to the use of the Okapi-BM25 algorithm. The results from several search engines including Bing, Google, and Yahoo! were extracted from several user search queries. When these results were stored within the database, along with other documents, the Fuzzy Web Information Retrieval System would find and rank all results from the other search engine relatively high in comparison to the other documents within the system. A benefit of using a fuzzy information retrieval system is that the document relevance to a query is given alongside the results notifying the user of document relevancy to their query entered.

5 Conclusion

Web information retrieval is an area open for many research opportunities. The larger problems with web information retrieval—relevance, evaluation, and information needs—amongst others, are still important topics that require attention [7]. Fuzzy information retrieval has proven to be a suitable solution for many areas involving information retrieval that may have data that can be uncertain, such as the web.

The individual sections of the system were developed. This includes a crawler system that obeys standard internet etiquette rules [10], and an indexing application that stores all information retrieved from the web in the form of a standard inverted index. Each section entered its gathered data to the database.

The ranking algorithm of this application was driven mainly using a developed fuzzy inference engine and several relational matrices were populated using standard web document ranking algorithms [11, 12]. The thesaurus relational matrix, along with the final user search was evaluated using the fuzzy relational BK-products defined in [2] with the different fuzzy implication operators were evaluated based on the validity of the initial results.

The results gathered from using the system developed in this study were comparable to existing methods for information retrieval. The relevant documents were consistently

given higher fuzzy membership values to a user query, and therefore, ranked higher than documents that were not relevant to a user query.

The major issue with using fuzzy relational methods and fuzzy logic in information retrieval on the web is the magnitude of the terminology base, and the scope of the documents to search from on the web. The concepts in fuzzy information retrieval are deeply involved with the use of matrix computations. This implies expensive spatial and temporal calculations which may not always be feasible in certain computing environments. High performance computing is mandatory for larger volumes of terms and documents. Tests that are limited in size however, can still be performed and evaluated on machines with fewer resources and good result sets can still be generated.

References

[1] L.A. Zadeh. " Fuzzy Sets." Information and Control. 8(3): 338 – 353, Elsevier Science. 1965.

[2] W. Bandler and L.J. Kohout. "Semantics of Implication Operators and Fuzzy Relational Products." International Journal of Man-Machine Studies, 12(1): 89 – 116, Elsevier Science, 1980.

[3] L.J. Kohout and W. Bandler. "Relational-Product Architectures for Information Processing." *Information Science,* 37(1-3): 25 – 37, Elsevier Science, December 1985.

[4] L.J. Kohout, E. Keravnou and W. Bandler. "Automatic documentary information retrieval by means of fuzzy relational products." B.R. Gaines, L.A. Zadeh, H.J. Zimmermann, Eds. , Fuzzy Sets in Decision Analysis. pages 383 – 404, North-Holland, Amsterdam, 1984.

[5] M. De Kunder. "The Size of the World Wide Web (The Internet)." http://www.worldwidewebsize.com/. WorldWideWebSize.com. Web. 25 Oct. 2011.

[6] M. Kobayashi and K. Takeda. "Information Retrieval on the Web." ACM Computing Surveys, 32(2): 144 – 173, June 2000.

[7] W. B. Croft, D. Metzler and T. Strohman. "Search Engines: Information Retrieval in Practice." Boston: Addison-Wesley. 2010.

[8] E. Liddy. "How a Search Engine Works." Searcher. 9(5): 38 – 45, May 2001.

[9] Z. Liu "Information Retrieval Using Relevance Feedback for the Mobile Internet." Thesis, University of North Dakota, May 2006.

[10] "The Web Robots Pages." http://www.robotstxt.org/**.** Web. 20 Apr. 2011.

[11] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. "Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC-3)." pages 109 – 126. Gaithersburg, MD, NIST, 1995.

[12] S.E. Robertson. "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF." Journal of Documentation. 60(5): 503 – 520, 2004.

[13] L. Yue. "Information Retrieval System Using Fuzzy Logic." Thesis, Florida State University, 1998.