Fuzzy Retrieval for Software Reuse

Dr. Erin Colvin
School of Computer Science
Colorado Technical University
4435 Chestnut Street
Colorado Springs, CO 80907
dr.erincolvin@yahoo.com

Dr. Donald Kraft
School of Computer Science
Colorado Technical University
4435 Chestnut Street
Colorado Springs, CO 80907
dkraft@coloradotech.edu

ABSTRACT

Finding software for reuse is a problem that programmers face. To reuse code that has been proven to work can increase any programmer’s productivity, benefit corporate productivity, and also increase the stability of software programs. This paper shows that fuzzy retrieval has an improved retrieval performance over typical Boolean retrieval. Various methods of fuzzy information retrieval implementation and their use for software reuse will be examined. A deeper explanation of the fundamentals of designing a fuzzy information retrieval system for software reuse will be examined. Future research options and necessary data storage systems are explored.

Keywords
Information retrieval, fuzzy retrieval, Boolean logic, software reuse

INTRODUCTION

Using term weights, we have developed a fuzzy search methodology, implementing the MMM, P-norm and Paice algorithms. This study is a replication of the 1991 study by Maarek using the same data corpus (UNIX man pages to represent a library of in-house software) but using fuzzy logic to measure similarity. These should result in a better matched list of returned values over the standard Boolean logic method. Looking at the degree of membership, or the fuzzy IR method, will yield a higher rate of successful match over the Boolean IR method. A higher success rate in returned software components should result in a higher reuse rate. This study is also the first to test extended Boolean functions vs. fuzzy IR with actual data.

This study has two goals, to prove that using a membership function is the best way to search and that using a membership function is the best way to search for software.

THE STUDY

Term frequency and inverse document frequency are calculated to determine the most frequent words in a document and which document contains a word the most frequently (Fox & Sharan, 1986). This indexing value helps search algorithms quickly look up a term and find the correlating documents that contain that term. The tfidf value will be used as the term weights in this research.

Using the term frequency calculation (tfidf) as the term weight, the similarity function in the MMM is calculated by:

$$SIM_{or} = \max\{\text{tfidf of all queried terms}\} + C_{or} \times \min\{\text{tfidf of all queried terms}\}$$

$$SIM_{and} = \min\{\text{tfidf of all queried terms}\} + C_{and} \times \max\{\text{tfidf of all queried terms}\}$$

where Q is the query, D is the document with index-term weights tfidf, and the C values are coefficients for “softness”, Prieto-Diaz says “since we would like to give the maximum of the document weights more importance while considering an or query and the minimum more importance while considering an and query” and usually $C_{and}$ is just $1 - C_{or}$ and $C_{or}$ is calculated as $1 - C_{or}$ (1991, p. 396). It was found that $C_{and} = 0.4$, $C_{or} = 0.6$, $C_{and} = 0.3$, $C_{or} = 0.7$.

The Paice model was proposed by Paice in 1984 and is also based on the fuzzy set theory. Similar to the MMM model, the Paice model looks at the weighted indexes in the document but doesn’t stop at the min and the max like the MMM model. The Paice similarity considers all of the
weights of the document. The Paice similarity is calculated by

\[ \text{SIM}(Q,D) = \sum_{(i=1)}^{n} \sum_{(j=1)}^{r} \frac{r^{(i-j)}}{\sum_{(i=1)}^{n} \sum_{(j=1)}^{r} r^{(i-j)}} \]

where \( n \) = number of queries, \( Q \) is the query and \( D \) is the tfidf for the document. For an OR query, \( D = (A_1 \text{ or } A_2 \text{ or } \ldots \text{ or } A_n) \), for an AND query, \( D = (A_1 \text{ and } A_2 \text{ and } \ldots \text{ and } A_n) \), where \( A_1 \) is the tfidf for query term 1, and so forth.

The P-norm model adds another angle to the Paice model by considering the weight of the query as well as the weights of the documents (Frakes & Baeza-Yates, 1992). In the research it has been found that \( p = 2 \) gives good results, for this research, the weight used will be 2. The P-norm similarity for an OR’d query is

\[ \text{SIM}(Q_{\text{orp}}, D) = \left( \sum_{(i=1)}^{n} a_i^{p} d_i^{p} + \sum_{(i=1)}^{n} a_i^{p} d_i^{p} + \ldots + a_n^{p} d_n^{p} \right)^{1/p} \]

where \( Q \) is the query, \( D \) is the document, \( a \) is the term weight, \( d \) is the document weight, \( p \) is set to 2, and \( A \) is the term for which the document weight is corresponding.

For an AND’d query the similarity is:

\[ \text{SIM}(Q_{\text{and}}, D) = \left( \sum_{(i=1)}^{n} (a_i^{p} d_i^{p} + a_i^{p} d_i^{p} + \ldots + a_n^{p} d_n^{p}) \right)^{1/p} \]

With a \( p \) value greater than 1 the computational toll is high, but to get a better result, computational expense is sometimes overlooked.

Based on the literature, there are no previous studies that looked at how effective a fuzzy approach (MMM, Paice, or P-norm) is for information retrieval of software. For this research, the MMM, Paice, Boolean and P-norm model were used to search a document for software components using the tfidf value as weights this was the first time fuzzy IR was compared to Boolean IR using actual software data.

**MEASURES OF PERFORMANCE**

The most well-known measure in the industry is recall and precision. Recall = |A \cap B|/|A| and precision = |A \cap B|/|B|, where A is the number of relevant documents and B is the number of retrieved documents.

This study also used the mean average precision (or MAP) to calculate the overall effectiveness of the system. This value was also used as a comparison measure (Manning, Raghavan, & Schutze, 2008). MAP is defined as:

\[ \text{AP}=\left(\sum_{(r=1)}^{N} (P(r) \times \text{rel}(r))/\text{NR}\right) \]

\[ \text{Mean Average Precision} = \frac{\text{AP}}{Q} \]

Where AP is the Average Precision and NR is the number of relevant documents. \( Q \) is the total number of queries, \( N \) is the number of retrieved documents, \( r \) is the rank in the sequence of retrieved documents, \( P(r) \) is the precision at rank \( r \), \( \text{rel}(r) \) is 1 if the item at rank \( r \) is relevant and 0 if it is not.

<table>
<thead>
<tr>
<th>Search Method</th>
<th>Paice</th>
<th>MMM</th>
<th>P-Norm</th>
<th>Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.575</td>
<td>0.541</td>
<td>0.541</td>
<td>0.40435</td>
</tr>
<tr>
<td>Ranking</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1. Ranking of results of search performance

**DATA RESULTS**

Based on these results, there is a 27.6% increase in the Paice model over the Boolean model and a 25.25% increase in the MMM and P-Norm model over the Boolean model. These results support the hypothesis that the fuzzy models will deliver a better search result over a Boolean model. This means that using this model, and with the right interface, an application can be made to search the UNIX man pages or other software that is designed similar to the UNIX man pages. Although that would be pretty useless as they are already searchable through the command prompt, it does mean this search algorithm can be used to further study other fuzzy methods, and how they would do with actual data. Other sources of data can be interchanged, such as GitHub, depending on the data needed for the application.

**FUTURE WORK**

There are many other areas for future research when it comes to information retrieval. This study did not consider time to execute or other measures of efficiency, but that would be an area that could be studied. Most of the literature discussed includes the similarity measure of time when discussing the success or failure of a new search algorithm, but that was not researched in this study.

The weight and ranking of an information retrieval system is another area that can be improved with future research. The basic concept of a ranking system is to assign a number or score to a document that matches the user’s query. This was done in this research using the fuzzy similarity measures compared to the Boolean score. Usunier, Buffoni, and Gallinari (2009) say that only a few top documents that match the query are really relevant to the user and those documents should have the highest precision and be at the top of the list, which they usually are not, so they devised a new ranking based on the Ordered Weighted Average (OWA).

One more area for future research that came up in doing this study is the idea of relevance. Relevance in an information retrieval system can come from three different areas – user-centered, systems-centered, or cognitive and is being looked at as the score that is compared in a study by Gupta, Saini, and Saxena. (2015). Deciding what is relevant to a user is difficult and has forced most IR systems to
create a network or mapping of information to related terms.

Crestani and Lalmas look at logic in an IR system and look at the relevance of documents as being true if a document meets a user’s needs, and if not the document is irrelevant (2001). This logical approach to relevance is the basis for their logical IR system, and to not sound like a straight Boolean IR system, they incorporate other theories to devise a new IR system that they compare to traditional IR systems. These new theories applied to this IR system would be a great area to expand for future research.

Another area that should be explored further is relevance feedback. Returning to the user a list of returned matches based on the searches results and letting the user then decide which of those results are relevant and then continuing to search based on the new results of the user. This idea of fine-tuning the results list is one that is currently being used by Google, and other popular search engines today.

Another area for future work is to replace the data corpus with other software libraries to see how the data would perform. Applying the results to an actual software library would be the ultimate test to determine if in fact this test does improve software retrieval as proved in this study.

CONCLUSION

There are many applications where an IR system can be effective. Moreover, the most accurate and most effective IR method is something that can benefit many. As we can see in this study, the fuzzy logic methods were an improvement over the Boolean search results. Because of this, more research can be done to find the most efficient system. There are many future routes where this research can be expanded, from query improvement to semantic vector space analysis to relevance feedback but it’s safe to say the fuzzy retrieval was a success over the Boolean model when actual data was applied and studied.

REFERENCES


