ESBIR: Extended Semantic Based Information Retrieval Algorithm using Synonyms and Antonyms

1 R. Thamarai Selvi, 2 Dr. E. George Dharma Prakash Raj

1 Asstt. Prof., Department of Computer Applications, Bishop Heber College, Tiruchirappalli-620017, INDIA.
2 Asso. Prof., School of Computer Sci., Enng.,& Technology, Bharathidasan University, Tiruchirappalli-620023, INDIA.

thams_shakthi@yahoo.co.in, georgeprakashraj@yahoo.com

ABSTRACT

Information Retrieval can be defined as the activity of obtaining information or data relevant to information needs from a collection of information resources. In general, resources are available in huge, vast and unstructured data formats. From the available resources, only a part or some is wanted by the users. Documents that satisfy the given query in the judgment of the user are said to be relevant. The documents that are not belonging to the given topic are said to be non-relevant. An IR engine uses the query to classify the documents in a given collection and returns a subset of documents that satisfy some classification criterion to the user. Sometimes the relevant documents may not contain the keywords specified in the query. The lack of the given term in a document does not necessarily mean that the document is not relevant because more than one term can be semantically similar although they are lexicographically different. In this paper a new algorithm, Extended Semantic Based Boolean Information Retrieval using Synonyms and Antonyms (ESBIR), is proposed to retrieve the documents with semantically similar terms to enhance the performance of Information Retrieval System by improving the recall and precision.

Keywords: Information Retrieval, Semantic, Word Net, Antonyms, Precision, Recall.

1. INTRODUCTION

Nowadays, information is stored in the database or repositories, which can be fetched or retrieved by other individuals. In order to have the ultimate use of the stored information it is necessary to retrieve the stored information more efficiently. The system of Information Retrieval (IR) deals with storing, maintaining and searching of information. The information that is retrieved should be relevant to the need of the user. It should be efficient and accurate. Information available on the internet is retrieved by different users at different times with different speed. Increasing the speed of information retrieval process can satisfy the user quickly. Such information retrieval process is not an easy task. Information retrieval is a major process which needs more attention to undertake all the above said. Information Retrieval can be defined as the activity of obtaining information or data relevant to an information needs from a collection of information resources [1].

IR provides the required information according to the user query within a collection of data. The Corpora that are available on the internet and information stored in the databases of libraries and companies are huge in recent years. This leads to the difficulty in searching of information needed from the vast data that exists. The need to efficiently organize, search and manage textual corpora for knowledge has brought a new interest in the process of information retrieval (IR) and data mining strategies. In IR system, the two important factors that directly affect the efficiency of the retrieval results is the approach to represent text and the measure to evaluate the similarity between query and documents. The task of ad hoc information retrieval is, finding documents within a corpus that are relevant to information need specified using a query. The main idea is to locate documents that contain terms that the users specify in queries. The lack of the given term in two documents does not necessarily mean that the documents are not related. Sometimes the documents may contain the opposite words (antonyms) of the given words preceded by the words like ‘not’, ‘im’, ‘un’, ‘ir’ etc. For example, the word ‘good’ has the meaning ‘not bad’. So, the document contains the word ‘not bad’ is also a relevant document for the user. Retrieval, by classical information retrieval models (e.g., Vector Space, Probabilistic, Boolean) [2] are based on lexicographic term matching. Therefore, these methods do not retrieve documents with semantically similar terms. In this paper a new algorithm is proposed to retrieve the semantically relevant documents with the use of Word Net database, antonyms and stemming algorithm Word Net is an on-line lexical reference system developed at Princeton University. It attempts to model the lexical knowledge of a native speaker of English.

The rest of this paper is organized as follows. Section 2 describes previous work in Semantic Based Boolean information model using Word Net. Section 3 defines the task of proposed work, its framework and algorithm along with the steps used. Section 4 contains the experimental results and discussion. And finally, the conclusion and future work is shown in Section 5.

2. RELATED WORK

The significance of Boolean Information Retrieval (BIR) has been revealed in many retrieval systems because of its simplicity [3]. Most of the commercial IR systems use this Boolean model to predict that each document is either relevant or not relevant [4]. For a number of reasons, both historic and technical, Boolean queries are particularly common in professional search. The number of studies over the years have shown that keyword queries are often significantly more
effective [5,6,7]. Boolean queries [8] however, are easy for information professionals to manipulate and are essentially self-documenting in that they define precisely the set of documents that are retrieved.

The semantic retrieval [9] approach is used to discover semantically similar terms using Word Net. In many works, Word Net is used to identify similar concepts that correspond to document words. In most cases morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR application. In linguistic morphology, stemming is the process for reducing inflected words to their stem, base or root form. The Porter stemmer [10] is a context sensitive suffix removal algorithm. Removing suffixes by is an operation which is especially useful in the field of IR. Word Net expansion technique was used [10] over a collection with minimal textual information.

Philip resniket. al. [11] annotated the Biblical text to create the aligned corpus such as Bible for linguistic research which also includes the automatic creation and evaluation of translation lexicons and similar tagged text. It has the feature of parallel translations over huge number of languages. It also represents the comparison with dictionary and corpus resources for modern English. Thus, it makes the Bible a multilingual corpus which considered to be a unique resource for linguistic research.

R. Thamarai Selvi et. al. [12] proposed an algorithm based on Boolean Information Retrieval (BIR) that has the significance of its simplicity. In this proposal, Boolean model is used to predict whether each document is relevant or not. It refers an online lexical reference called Word Net to find the semantically similar terms. Stemming is the process which is used for reducing inflected words to their stem, root or base form.

3. PROPOSED WORK

Several search engines have been developed to find information in repositories containing large amounts of unstructured form of text data. Ad hoc information retrieval is finding documents within a corpus that are relevant to information need specified using a query. Sometimes, the document may contain the opposite words of the given words in the query, preceded by the words like ‘not’, ‘im’, ‘un’, ‘ir’ etc. For example, the word ‘irrelevant’ has the meaning ‘not relevant’. Hence, the document contains the word ‘not relevant’ is also a relevant document for the user. In this proposed algorithm called “Extended Semantic Based Information Retrieval Algorithm” information retrieval is done in an effective way by increasing the number of relevant documents, improving the precision and recall, and reducing the time taken for retrieving information using user feedback. Fig. 1 shows the complete architecture of the proposed algorithm.

Today search engine is very helpful for retrieving a large amount of information with a single word or text as the query. It has to find the related information from all available repositories or databases, which contain huge amount of information. This proposed algorithm performs the information retrieval by using the term in the user query. The information given by the user in the form of text, words or sentences is considered as the query. Each query is composed of terms.

3.1 Esbir Framework

This Extended Semantic based Boolean Information Retrieval Algorithm using synonyms and antonyms provides a novel perspective for approaching the task of ad hoc retrieval. In ad-hoc information retrieval, the user formulates any number of arbitrary queries and applies them to a fixed collection. The task of ad hoc information retrieval, finding documents within in a corpus that are relevant to an information need specified using a query. The framework for ESBIR is given below in the Figure 1.

![Fig 1: Esbir Framework](image-url)
3.2 ESBIR Algorithm

This work targets to develop a system to address the information retrieval for a static data set and aims to provide documents from the collection that are relevant to an arbitrary user information need. An information need is the topic about which the user desires to know more. ESBIR retrieves documents from the document set by finding synonyms and antonyms of the given term using Word Net database to find more similar documents. The proposed ESBIR algorithm is given below.

Algorithm: ESBIR

```
Input : Documents collection
Output : Relevant documents and index

Algorithm:
Step 1:
  Preprocessing the documents
Step 2:
  Enter the query term tm
  Find the semantically similar terms s and assign in T
Step 3:
  For each si, in T
    If si is prefixed with “im” or “ir” or “un”
      Remove the above said prefix from si
      Find the antonyms of si and assign to PTL
    else
      Find the antonyms of si, add the suitable prefix “not” or “im” or “ir” or “un” and assign to PTL.
Step 4:
  User selects the words from PTL and assign to UPTL
Step 5:
  Find the stemmed word st,
  for each si, in UPTL and assign to SPTL
Step 6:
  for each si, in UPTL
    find the document d and put them in Di
    create index for the term
Step 7:
  for each di, in Di
    display the documents di
```

The steps given in this algorithm are explained below.

**Step 1: Preprocessing**

The documents are preprocessed and the information is stored in my sql database. This step is performed to retrieve the individual documents from the corpus and store them in the database. The keywords are also extracted from the document by eliminating the stop words and stored them in the data base.

**Step 2: Retrieving Synsets from Word Net**

In the second step, we retrieve the synsets from the Word Net data base for the given keyword. These set of words are stored in a data structure array. Word Net is an on-line lexical reference system developed at Princeton University. Word Net attempts to model the lexical knowledge of a native speaker of English. Word Net can also be seen as an ontology for natural language terms.

**Step 3: Adding antonyms**

Each term in the synsetis checked whether the term is prefixed with “im”, “ir” or “un”. If so, these prefixes will be removed and the antonyms of the words will be found then add “not” to these terms and stored in the PTL (Process Term List). Otherwise, the antonyms of each word in T is found and then the suitable prefix like “im”, “ir”, “un” is added before the antonyms and added in the PTL.

**Step 4: Selecting the set of words**

In this step, the user selects some set of words from PTL and assigns to the User Processed Term List (UPTL), because the PTL may contain many words in which the user may not have interest to search. In this case, this algorithm allows the user to feed back the necessary terms for searching.

**Step 5: Stemming Process**

In this fifth step, the stemming process is done on the UPTL for each word. The Porter Stemming Algorithm implemented in java performs this process. Porter’s stemmer is more compact and easy to use than Lovins [12]. These words are stored in a vector. These stemmed words will be used to retrieve the documents from the data base.

The Porter stemming algorithm is a process for removing the commoner morphological and in flexional endings from words in English. Stemming algorithms are used in many types of language processing, text analysis systems, information retrieval and database search systems. Its main use is for term normalization process that is usually done when setting up Information Retrieval systems. Word stemming is an important feature supported by present day indexing and search systems. The idea is to improve recall by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching.

**Step 6: Retrieving Documents**

This algorithm is tested on the Bible (Kings James Version) data base. This collection contains 66 books, 1189 chapters, 31,102 verses and 7,882,80 words. The relevant documents are retrieved from this data base for each word in the SPTL and stored in D_i and create the index for the query term(t_m).

**Step 6: Display the Documents**

All the documents that are retrieved by this algorithm are displayed to the user.

4. EXPERIMENTATION AND ANALYSIS

4.1 Experimental Setup

This ESBIR algorithm to improve the precision and recall of the information retrieval is implemented using Java programming language and tested on Bible text. This text contains many small documents called verses. These verses are converted into individual documents and stored in the MySql database. The dimension of the documents is reduced using the stop word elimination and all the key terms are stored in the database.
The synsets and the antonyms are extracted from WordNet using Java APIs and stored in vectors to test the ESBIR algorithm. Each term in the vector is stemmed to root words and stored in another vector. For example, the set of words for the word ‘accept’ are consent, receive, willing, not reject, not refuse, and not decline. These words are stemmed to their root word. After stemming process, the documents are retrieved from the database for all the root words. For example, the synsets for the word “impossible” are “not capable”, “unacceptable”, “unimaginable”, “not possible”, “not attainable”, “not acceptable”. These words are stemmed to their root word. After stemming process, the documents are retrieved from the database for all the root words.

4.2 Evaluation of Sbir and Ufsbir

The number of documents retrieved by ESBIR is larger than the documents retrieved by using the UFSBIR algorithm.

Table 1 shows the number of documents that are retrieved by using UFSBIR and ESBIR.

<table>
<thead>
<tr>
<th>Query</th>
<th>UFSBIR</th>
<th>ESBIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad</td>
<td>604</td>
<td>629</td>
</tr>
<tr>
<td>impossible</td>
<td>184</td>
<td>193</td>
</tr>
<tr>
<td>uncover</td>
<td>327</td>
<td>332</td>
</tr>
<tr>
<td>close</td>
<td>210</td>
<td>223</td>
</tr>
<tr>
<td>accept</td>
<td>161</td>
<td>169</td>
</tr>
<tr>
<td>reject</td>
<td>108</td>
<td>149</td>
</tr>
<tr>
<td>weak</td>
<td>79</td>
<td>89</td>
</tr>
<tr>
<td>forget</td>
<td>74</td>
<td>82</td>
</tr>
<tr>
<td>remember</td>
<td>169</td>
<td>180</td>
</tr>
<tr>
<td>wrong</td>
<td>34</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 1: Number of Documents Retrieved

In information retrieval, Precision and Recall are the basic measures used in evaluating search strategies. Recall is defined as the number of relevant documents retrieved divided by the total number of existing relevant documents and precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved by that search. Table 2 shows the systematic and traditional notations of confusion matrix.

<table>
<thead>
<tr>
<th></th>
<th>RELEVANT</th>
<th>NOT RELEVANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>Not Retrieved</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix

Here,

TP = True Positive (Correct Result)
FN = False Negative (Missing Result)
FP = False Positive (Unexpected Result)
TN = True Negative (Correct absence of Result)

Recall = TP / (TP + FN)
Precision = TP / (TP + FP)

The values obtained by the two algorithms UFSBIR and ESBIR are entered in the confusion matrix for different keywords and the precision and recall values are calculated. The Table 3 shows the Precision and Recall of the two algorithms.

Table 3: Precision and recall of sbir and ufsbir

<table>
<thead>
<tr>
<th>Query</th>
<th>UFSBIR</th>
<th>ESBIR</th>
<th>UFSBIR</th>
<th>ESBIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad</td>
<td>0.9738</td>
<td>0.9749</td>
<td>0.9252</td>
<td>0.9668</td>
</tr>
<tr>
<td>impossible</td>
<td>0.8721</td>
<td>0.8785</td>
<td>0.7772</td>
<td>0.8238</td>
</tr>
<tr>
<td>uncover</td>
<td>0.9677</td>
<td>0.9636</td>
<td>0.9036</td>
<td>0.9187</td>
</tr>
<tr>
<td>close</td>
<td>0.9091</td>
<td>0.9147</td>
<td>0.8072</td>
<td>0.8655</td>
</tr>
<tr>
<td>accept</td>
<td>0.8765</td>
<td>0.8912</td>
<td>0.6711</td>
<td>0.8792</td>
</tr>
<tr>
<td>reject</td>
<td>0.8571</td>
<td>0.875</td>
<td>0.6742</td>
<td>0.7865</td>
</tr>
<tr>
<td>weak</td>
<td>0.8857</td>
<td>0.8974</td>
<td>0.7561</td>
<td>0.8537</td>
</tr>
<tr>
<td>remember</td>
<td>0.9186</td>
<td>0.9235</td>
<td>0.8778</td>
<td>0.9389</td>
</tr>
<tr>
<td>wrong</td>
<td>0.8108</td>
<td>0.9348</td>
<td>0.6</td>
<td>0.8776</td>
</tr>
</tbody>
</table>

The performance of UFSBIR and ESBIR is represented as Precision graph and Recall graph in Fig 3 and Fig 4.

Fig 3: Precision Graph
Fig 4: Recall Graph

The graphs show that precision and recall of the proposed algorithm ESBIR is increased compared to UFSBIR algorithm for the given queries.

5. CONCLUSION

In this paper, a new algorithm called ESBIR is proposed to enhance the performance of Semantic Based Boolean Information Model. Since the proposed algorithm considers the synonyms as well as the antonyms of the search word, it creates the possibility of retrieving more number of documents. This has been proved through the experiments and the results. The precision and recall values that are calculated through the experiments confirm the improved performance of ESBIR over UFSBIR.

REFERENCES


