CHAPTER 3

GENERIC SEMANTIC SEARCH USING NLP TECHNIQUES AND SCBR MODEL

3.1 PREAMBLE

The generic semantic search is designed for the users who do not have domain knowledge with regard to their search queries. Since the search requesters lack sufficient knowledge regarding their search queries, these queries could contain ambiguities, misleading, unmatched kind of service information, and the wrong search queries. Consequently, the task of this system is to help the users to disambiguate those queries and denote their query intentions with search domain knowledge, in order to precisely find the search information which can best apt to their query objectives.

The rest of chapter is organized as follows: section 3.2 describes the problem formulation and objectives. Section 3.3 explains the workflow of the proposed generic semantic search technique. Section 3.4 evaluates the performance analysis and discussion and section 3.5 presents summary of the chapter.

3.2 PROBLEM FORMULATION AND OBJECTIVES

There is no proposal for assisting search requester without relevant knowledge about their queries to amend incorrect or incomplete queries. Even though, the search result is of poor performance in recall, precision value.
This research focuses to develop a reliable and an efficient generic semantic search technique to retrieve accurate results for the user’s complex query with respect to the educational domain. It even bears the human error in typing, and suggests the expected word to search for. It also aims at retrieving the same result for synonymous words and prevents the appearance of irrelevant search results. It provides complete details for the query about the required data with the correct URL and metadata. The details are generated with the help of ontology and relations among classes, entities, individuals, data type properties, object properties and restrictions are also created. Hence now the user can query upon the information stored within the Ontology Knowledge Base (OKB).

3.3 PROPOSED GENERIC SEMANTIC SEARCH TECHNIQUE

The generic semantic search technique of the semantic search system is highlighted in Figure 3.1. The technique consists of Natural Language Processing (NLP) technique, Semantic Case Based Reasoning Model (SCBR) and Ontology Knowledge Base (OKB).

Figure 3.1 Generic semantic search technique
Ontology knowledge base is to store ontology and metadata, NLP technique for spell check and to find synonyms, SCBR model for query concept matching process. The proposed techniques are contributing best performance results in precision, mean average precision and recall.

3.3.1 Process Flow Diagram

The proposed mechanism is explained through the process flow diagram as shown in Figure 3.2.

![Figure 3.2 Process flow diagram of generic semantic search](image)

The user enter query in the search interface, sends each query term to the spell checker for correcting the user query for incorrect or spelling mistakes, the spell checker provides suggestions and correct spelling mistakes,
then it passes each of the query words to synonymer such as WordNet API. If one query term can be retrieved from the synonymer, the synonymer returns its synonyms; otherwise, the query term is filtered. After the process being completed, the synonymer sends the query terms and their synonyms to the query concept matching task. The query concept matching task use the SCBR model to compute the similarity values between the query term and ontology concepts stored in the ontology knowledge base and if the query terms matched in OKB concept then retrieved relevant information is given to the user. Once the user selects or accepts a result, all its semantically relevant metadata will be displayed from the ontology knowledge base.

### 3.3.2 Ontology Knowledge Base

The ontology knowledge base is developed to store ontology and metadata for the educational domain dataset by using Protégé OWL. It consist two main components ontologies base and metadata base in which semantically related ontological concepts and metadata are linked by referencing their URL to one another. It has two rules contained in the semantic relationship; the first one is a concept which may semantically relate to arbitrary metadata and the second one is a metadata which may also semantically relate to arbitrary concepts. The ontology knowledge base is shown in Figure 3.3.

![Ontology Knowledge Base](image)

*Figure 3.3 Ontology knowledge base*
Ontology Base is designed to store ontological concepts. Ontology is denoted as conceptualization of the data, in which identified by concept name, concept description and linked metadata. The ontology is an amalgamation of ontology name and a tuple. The elements of the tuple can be complex elements and the concept name can be used to uniquely identify a search query. The data description refers to the definitional descriptions of a query; the normal form of a data description is a set of words like noun, adjective or adverb. Concepts have many query descriptions. The adventure of setting the property of service description is to compute the semantic similarity values between concepts and queries, linked metadata refers to the URL of semantically related metadata to a concept. The ontology is the definition of the query concept in the root of the term concept hierarchy. As leaf concepts all other concepts in this hierarchy automatically inherit its properties.

Metadata Base is designed to store query concept processed data. The purpose of metadata is to bring out meaningful information with regard to the real environment. The metadata defined as linked concepts, concept name, address, contact details, and metadata descriptions. The linked concepts refer to the URL of semantically related to concepts to the metadata. Concept name refers to the name of the college or institution. Address refers to the address where it can be located. Contact details refer to the information regarding telephone number, fax number, website and so on. Description refers to the detailed text description regarding the content of a search query. This can be used for matching with concepts.

This research made use of educational ontology for querying upon the desired event, the required components to build up the ontology such as classes, instances and relationships are being created. The classes created are college, college type, school, university, district, etc., the subclasses created
within the college are engineering, arts and science, law, medical, polytechnic, institutions etc., with regard to the metadata. The background knowledge for the ontology is obtained from the Linked Open Vocabulary dataset (LOV) and the following (http://www.studyguideindia.com), (http://www.india.gov.in/education), (http://www.tn.gov.in/schooleducation), (http://en.wikipedia.org/wiki/list_of_engineering_college_ranking) web sites. The properties of the classes are created. Properties are of two types:

- Data type Property: It is being used to set properties, enhancing the existence of an individual.
- Object Property: It is used to create a relationship between two different class individuals.

![Figure 3.4 Sample educational ontology screen shot](image)

In Figure 3.4 shows the sample ontology screen shot. The data type properties created is about college, contact, location or address and URL. The object properties created are type of colleges, locations, the name of
institutions etc., The individuals for each OWL classes are created for engineering colleges, arts and science colleges, schools etc.,

3.3.3 NLP Techniques

This research utilized NLP techniques spellchecker and synonymer (WordNet API) for correcting wrong search queries and finding synonyms.

3.3.3.1 Spell checker

Spell checker aims to provide a preprocessor for an information retrieval system allowing the user’s query to be checked against a dictionary and corrected any spelling errors to prevent wasted efforts. Information retrieval, searching is computationally intensive so much, if to prevent futile searches can minimize computational cost. Spelling errors are abundant in the queries generated by the users and can easily defeat search and retrieval operations in IR system if not detected. In our work, query concept matching process is required to detect errors in user’s queries. The error checking would prevent fruitless searching for misspelt words which wastes both computational processing and user time and would make the system more robust to user errors.

In this IR system, the user supplies a simple query, as a list of the required search words. Each word is matched against a list of all terms present in the text corpus by the spell checker. If the word is available, then the user can either elect to just query on that term or search for similar terms using the spell checker, where the terms have the same word branch as the query term. If the word is not available, then the mechanism assumes a spelling error and the spell checker suggests a list of alternative spellings from which the user can select word to be used in the query. The spell check algorithm is shown in
Figure 3.5, it is simple and flexible and it is main aim is to have high recall possible at the expensive or precision.

<table>
<thead>
<tr>
<th>Procedure Spellcheck <em>(query q_word, dict_word, corrected_query cq)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td>Query</td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
<tr>
<td>Correct Query</td>
</tr>
<tr>
<td><strong>Begin</strong></td>
</tr>
<tr>
<td><strong>While</strong> not end of Query <strong>do:</strong></td>
</tr>
<tr>
<td>Read next word from query: <em>q_word</em></td>
</tr>
<tr>
<td>Select a word from CUSTDICT: <em>dict_word</em></td>
</tr>
<tr>
<td><strong>If</strong> <em>q_word ≠ dict_word</em> <strong>of all words in CUSTDICT</strong> <strong>then</strong></td>
</tr>
<tr>
<td>Assume spell mistake and suggest list of possible alternative word to <em>q_word</em></td>
</tr>
<tr>
<td><strong>Else</strong></td>
</tr>
<tr>
<td>Assume correct word and go to synonymer to find synonyms</td>
</tr>
<tr>
<td><strong>End if</strong></td>
</tr>
<tr>
<td><em>cq = q_word</em></td>
</tr>
<tr>
<td><strong>End</strong></td>
</tr>
</tbody>
</table>

Every *q_word*, is compared to a single *dict_word*

Where

*dict_word = min { w: w≥t_word and w ∈ CUSTDICT}*

If *q_word = dict_word*, then *q_word* is correct and already in CUSTDICT. Otherwise, since it is possible that *q_word* is a correct word not yet present in CUSTDICT, the user is given the option to add *q_word* to CUST DICT. If that option is declined, *q_word* is assumed to be mistyped, therefore the method suggest list of possible alternative word to query.

**Figure 3.5 Spell-check algorithm**
The users enter incorrect words or spelling mistakes, the algorithm provides suggestions for unknown (misspell) words based on custom dictionary. The spell check algorithm has “Advanced Suggester”. The “Advanced Suggester” uses multiple dictionaries with different weights assigned to each dictionary and each word. It also supports multiple languages. It provides suggestions for unknown (misspell) words based on custom dictionary. The “Advanced Suggester” creates a list of preferred words and assigns a higher weight to the list, as a basic implementation “Suggester” it can serve as a spell checker. The basic implementation includes high speed suggestion engine based on fast edit-distance calculation algorithm (Kukich 1992) enhanced with Lawrence Philips Metaphone algorithm and private fuzzy-matching algorithm.

The spell checker may be used in some other application areas, for example: Improving communication between speech and language impaired individuals; historical data search, for over the centuries the English language and word spelling have progressed. Hence, performing an effective search on these documents require an awareness of spelling errors; Genetic research applications, for properties and behavior of molecules depend on the sequence of basic elements much the same way as words are sequences of the alphabets; data imputation, for responses to any queries of a questionnaire may be known to be in error because they fail one or more “edits”. The problem is to determine the true intended response; search on media databases, for recent growth in digital representation of music has led to the creation of large music databases. There is considerable interest in efficient retrieval techniques for such music collections.
3.3.3.2 Synonymer (Finding synonym using WordNet API)

After the spell checking, the WordNet API employed for finding synonyms for meaningful search is also known as “synonymer”. In the WordNet, the words and their relationships to each other are organized in a hierarchical manner similar to the taxonomies which may be found in the natural sciences. Words which are closely related to each other may be found in the same branch of the hierarchical tree. Each word belongs to a set of synonyms, also known as a “synset”. These “synsets” are the foundation upon which the WordNet database is constructed. Formally, a “synset” is a set of one or more synonymous words that may be substituted for each other in context without changing the overall meaning of the sentence in which they are contained. One word has numerous meanings or “word senses” appear in more than one “synset”. WordNet provides a polysemy count for each word which is used to track the number of “synsets” which contain the word.

Since different word types follow different grammatical rules, WordNet makes the distinction between four of the primary word types in the English language, which include nouns, verbs, adjectives, and adverbs. The noun category contains words which refer to entities, qualities, states, actions, or concepts, and can serve as the subject of a verb. Words classified as verbs may serve as the predicate of a sentence and describe an action, occurrence, or state of existence. Adjectives are words that may modify nouns. The final word classification stored in WordNet, the adverb, is similar to the adjective and contains words which modify word types other than nouns.

Importing WordNet: The JAWS is a (Java API for WordNet Searching) runtime library and it is adding to Java archive (.jar) file in the class path of the application which will be used WordNet. A statement which imports “edu.smu.tspell.wordnet” should be included in .java source files making use
of JAWS. The path to the WordNet database should also be specified by setting the system property “wordnet.data.dir” to the “dict” subdirectory of the root WordNet directory.

**Instantiating a WordNet Database:** The WordNet database class provides access to the information stored in the WordNet database and must be instantiated before use. A method, `get.File.Instance`, returns an implementation of the class that works with the local WordNet database and may be used when creating a new instance of the WordNet database class. Other than WordNet Database, another critical component of the JAWS API is the, “Synset” interface. This interface represents WordNet's collections of related words, or “synsets”. These “synsets” are stored as an array of word forms. Several overloaded methods of the WordNet database class known collectively as “getSynsets” can be used to retrieve “synsets” from the WordNet database by providing a starting word in the form of a string when the “getSynsets” is called. Instantiating a “synset”, the “getSynsets” method is used to populate the new instance of the Synset interface with WordNet information.

**Retrieving Synonyms:** The `getWordForms` method used to retrieve the individual groups of word forms for each “synset” stored as an element of this array, which may themselves be stored as arrays of strings containing all words similar to the original word.

### 3.3.4 Semantic Case Based Reasoning (SCBR) Model

Semantic Case Based Reasoning (SCBR) algorithm is proposed for query concept matching in semantic search to obtain efficient search results. It is an enhanced version of Extended Case Based Reasoning (ECBR) algorithm
(Dong et al 2011). The process flow diagram of SCBR model is shown in Figure 3.6.

![Figure 3.6 Process flow diagram of SCBR model](image)

The principle of the SCBR model is to seek the maximum similarity value between a user query and concept stored in OKB. If SCBR model finds exact data in the OKB for the given user query term, then it assigns a value as 1. If SCBR model can obtain relevant information for the given user query term, it assigns its similarity value as 0.5. Otherwise 0 will be assigned. The optimal threshold value is dynamically set between 0 to 1 and the performance of SCBR model is analyzed for each time with on the increment of value 0.1. A threshold value needs to filter irrelevant data and then attain the performance of concepts for each time variation of the threshold value. The query is then compared with metadata description.
property of each concept from the OKB. The maximum value between the query and data description properties of a concept is considered as the similarity value between the query and the concept.

The ECBR model does not propose a spell check method (Dong et al 2011). The SCBR model aims to provide a preprocessor for an information retrieval system allowing the user’s query to be checked against a dictionary and any spelling errors corrected to prevent unfruitful searching. The spell checker is designed such that it gets the query and finds the correct word which is passed to get the synonyms and finally the word with its synonyms is passed to the query concept matching model. The similarity values calculated based on SCBR model can be mathematically shown as:

\[
\text{sim}(q, c) = \max_{md_i \in c} \left( \sum_{k_{ih} \in md_i} \frac{f(s_{cih}) + f(q, k_{ih}) + fm(m, k_{ih})}{\sum md_i} \right) \tag{3.1}
\]

with,

\[
f(q, k_{ih}) = \begin{cases} 
1 & \text{if } \exists q_{kt}(s_{cih}) \left( \forall st, wt(k_{ih}) = wt(q_{kt}) \wedge (q_{kt} \in q) \right) \\
0 & \text{otherwise}
\end{cases} \tag{3.2}
\]

\[
fm(m, k_{ih}) = \begin{cases} 
0.5 & \text{if } \exists q_{kt}(s_{cih}) \left( \forall st, wt(k_{ih}) = wt(q_{kt}) \wedge (q_{kt} \in m) \right) \\
0 & \text{otherwise}
\end{cases} \tag{3.3}
\]

where, \(q\) is a processed query, \(c\) is a concept, \(md_i\) is a meta data descriptions property of concept \(c\), \(k_{ih}\) is a key involved in \(md_i\), \(\sum md_i\) is the sum of associated with \(md_i\), \(q_{kt}\) is the query key term involved in \(md_i\), \(s_t\) is the semantic term, \(w_t\) is a function that returns a weight associated with \(s_t\), \(m\) is a meaning of query, \(s_{cih}\) is spellchecker.
3.4 PERFORMANCE ANALYSIS AND DISCUSSION

The system evaluation is divided into two parts: 1) System implementations and functional testing to validate the proposed work. 2) Evaluating the employed mathematical model to test their performance. The test data are the educational domain collecting from LOV (Linked Open Vocabulary) and various websites.

Prototype Implementation and Functional Testing

In this section, the work implemented semantic search engine and run with different input queries relevant to educational domain. For example the user input as engineering college it retrieves the list of engineering colleges and when selected a specific college from the list gives its corresponding location, URL and contact followed by the mission of the institution. When the required district to which the query term is relevant, it displays results corresponding to the district. Though the user enters the words with the wrong spelling, it retrieves the output with the correct word from OKB. The result is being retrieved the same for synonymous words. The proposed work is being implemented with the available tools and environment as follows,

- Eclipse3.6.0: is an integrated development environment for developing applications in Java, and it is used as the platform to develop an educational search system.
- Java Development Kit: JDK1.7 (Java software Development Kit) is used for the implementations of this framework.
- Wamp server: Wamp server is used to connect the ontology file and Java code. The ontology file is placed in www directory.
- Protégé OWL: is ontology editor, which is used as the platform for developing education domain ontology in the knowledge base.
• Protégé-OWL API: Protégé-OWL is an open source Java library for the web ontology language and RDF(S). Here the Protégé-OWL API is used to load an OWL coded ontology.

• Jena: Jena is a Java framework for building semantic web applications and here is used to load RDF(S), SPARQL coded ontology.

• Java script and AJAX: is used for display the locations through Google maps.

• Custom dictionary of BasicSuggester: for spell checking the given input query.

• WordNet API for finding relevant data.

Evaluating the Mathematical Models for the Semantic Search

As described previously, there are four models used and compared to semantic search technique, which are SCBR, ECBR, VSM, and LSI model in this work concentrate on evaluating the four performance indicators of these models as follows.

Performance Indicators

The four performance indicators used in this experiment are precision, mean average precision, recall and f-measure as follows,

Precision is used to measure the preciseness of a search system (Baeza-Yates & Ribeiro-Neto 2011). In this experiment, Precision P is defined as the number of retrieved relevant data among the retrieved data.

\[
Precision \ P = \frac{\text{number of retrieved relevant data}}{\text{number of retrieved data}} \quad (3.4)
\]

In advance announce the definition of mean average precision, the data of average precision must be defined. Average precision stands the
average of precision values at each retrieval relevant data for a query, given that these data are ranked according to their computed similarity values. This indicator is used to measure how quickly and precisely a search engine works (Baeza-Yates & Ribeiro-Neto 2011).

\[
\text{Average precision}(Q) = \frac{\text{sum( precisions @ retrieved relevant data)}}{\text{number of retrieved relevant data}}
\]  

(3.5)

Mean average precision refers to the average of average precision values for a set of queries and can represented as.

\[
\text{Mean average precisions} = \frac{\sum_{i=1}^{n} \text{Average precision}(Q_i)}{n}
\]  

(3.6)

Recall is used to measure the effectiveness of a search system (Baeza-Yates & Ribeiro-Neto 2011). In this experiment, Recall R is defined as the number of retrieved relevant data to total number of relevant data in the knowledge base.

\[
\text{Recall} R = \frac{\text{number of retrieved relevant data}}{\text{number of relevant data}}
\]  

(3.7)

F-Measure combines precision and recall, in this research is used as an aggregated performance scale and users can specify the preferred on recall or precision by configuring different weights. When the F-Measure value reaches the highest, it means the integrated value between precision and recall reaches to the highest at the same time (Baeza-Yates & Ribeiro-Neto 2011).

\[
F – \text{Measure} = \frac{2PR}{(P + R)}
\]  

(3.8)
**System Evaluation Results:** To examine the performance of the SCBR model, compare with three IR models such as Extended Case Base Reasoning (ECBR) (Dong et al 2011), Vector Space Model (VSM) and Latent Semantic Indexing (LSI). The mechanism and algorithm concerning the model referred from (Baeza-Yates & Ribeiro-Neto 2011). Different queries are made to compare the performance of the system. All the parameter results are averaged by 100. These queries cover most of the general user requirements in the educational domain. A threshold values need to be configured to select the most similar concepts by filtering the concepts with the lower similarity values.

In addition to this, there is a need of finding an optimal threshold for each IR model. In order to achieve this, the similarity values between a query and the concepts are computed for different IR models and are compared.

A threshold needs to be selected for filtering the relatively dissimilar concepts to obtain the optimal performance for each model. Owing to the difference between each model, the optimal threshold value could be different. To choose the optimal threshold utilizes the f-measure as the primary scale. The threshold scope is configured between 0 to 1 with an increment of 0.1 at each time, then to evaluate with four information retrieval algorithms and to choose the optimal thresholds with the overall performance of the search process, based on the same set of queries.

In the experimental results, “The WordNet API features” can also be read as “NLP techniques” (spell checker and synonymer).
Table 3.1 presents testing results of SCBR model with WordNet and without WordNet features. It is observed that along with the increase of the threshold value (0 to 0.9) the precision results a sharp rise. Mean average precision tests for the quickness and precious of a search results. Recall tests for the accuracy. F-measure is an aggregated metrics of precision and recall.

Table 3.1 Testing results of SCBR model

<table>
<thead>
<tr>
<th>Optimal Threshold Value</th>
<th>P % W*</th>
<th>P % W</th>
<th>MAP % W*</th>
<th>MAP % W</th>
<th>R % W*</th>
<th>R % W</th>
<th>F-M % W*</th>
<th>F-M % W</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>32.98</td>
<td>17.72</td>
<td>82.98</td>
<td>72.48</td>
<td>80.50</td>
<td>70.85</td>
<td>46.79</td>
<td>28.35</td>
</tr>
<tr>
<td>&gt;0.1</td>
<td>43.98</td>
<td>17.72</td>
<td>83.98</td>
<td>72.48</td>
<td>78.17</td>
<td>70.85</td>
<td>56.29</td>
<td>28.35</td>
</tr>
<tr>
<td>&gt;0.2</td>
<td>45.21</td>
<td>23.63</td>
<td>85.21</td>
<td>72.66</td>
<td>77.17</td>
<td>70.50</td>
<td>57.01</td>
<td>35.40</td>
</tr>
<tr>
<td>&gt;0.3</td>
<td>58.22</td>
<td>26.92</td>
<td>88.22</td>
<td>72.95</td>
<td>76.41</td>
<td>70.16</td>
<td>66.08</td>
<td>38.90</td>
</tr>
<tr>
<td>&gt;0.4</td>
<td>61.18</td>
<td>29.65</td>
<td>91.18</td>
<td>81.52</td>
<td>75.34</td>
<td>60.27</td>
<td>67.52</td>
<td>39.76</td>
</tr>
<tr>
<td>&gt;0.5</td>
<td>72.39</td>
<td>29.56</td>
<td>92.39</td>
<td>81.67</td>
<td>63.35</td>
<td>60.02</td>
<td>67.56</td>
<td>39.63</td>
</tr>
<tr>
<td>&gt;0.6</td>
<td>84.34</td>
<td>71.54</td>
<td>95.35</td>
<td>90.33</td>
<td>52.29</td>
<td>40.74</td>
<td>64.55</td>
<td>51.93</td>
</tr>
<tr>
<td>&gt;0.7</td>
<td>96.26</td>
<td>80.02</td>
<td>97.26</td>
<td>90.66</td>
<td>49.37</td>
<td>37.13</td>
<td>65.26</td>
<td>50.72</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>97.37</td>
<td>80.96</td>
<td>98.37</td>
<td>90.69</td>
<td>48.53</td>
<td>36.62</td>
<td>64.77</td>
<td>50.43</td>
</tr>
<tr>
<td>&gt;0.9</td>
<td>98.52</td>
<td>80.98</td>
<td>99.53</td>
<td>90.69</td>
<td>47.98</td>
<td>36.62</td>
<td>64.53</td>
<td>50.45</td>
</tr>
</tbody>
</table>

where P % = Precision in %, MAP % = Mean Average Precision in %, R % = Recall in %, F-M % = F-Measure in %, W* = with WordNet, W = without WordNet.

Using WordNet, the highest value of precision increases from 32.98% to 98.52% at the threshold value 0 to 0.9, mean average precision value from 82.98% to 99.53% at the threshold value 0 to 0.9, in contrast, when the threshold value increases the recall ranges decreases from 80.50% to 47.98% at the threshold value 0 to 0.9. F-measure value ranges, peak is 67.56% of the threshold value in 0.5. In without WordNet features, the
precision, mean average precision, a recall and f-measure value is relatively low when compared with WordNet features. The ranges of precision 17.72% to 80.98% at the threshold value 0 to 0.9, mean average precision is 72.48% to 90.69% at the threshold value 0 to 0.9, recall is 70.85 % to 36.62% at the threshold value 0 to 0.9 and f-measure is 28.35% to 51.93% at the threshold value 0 to 0.9. Therefore the NLP features (synonymer and spell checker) increases the performance of our proposed work and it is gaining a high level of search performance.

Table 3.2 presents testing results of ECBR model with and without WordNet features. Using WordNet the precision values are 12.39% to 79.47%, mean average precision 70.98% to 90.69%, recall 75.26% to 36.65% and f-measure 21.28% to 51.05% at the threshold value 0 to 0.9 and without WordNet the values of precision 17.43% to 81%, mean average precision 72.48% to 90.67%, recall 70.85% to 36.62% and f-measure 28.36% to 50.45% at the threshold value 0 to 0.9 . It is observed that the WordNet API decreases the performance of ECBR.

**Table 3.2 Testing results of ECBR model**

<table>
<thead>
<tr>
<th>Optimal Threshold Value</th>
<th>P % W*</th>
<th>P % W</th>
<th>MAP % W*</th>
<th>MAP % W</th>
<th>R % W*</th>
<th>R % W</th>
<th>F-M % W*</th>
<th>F-M % W</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>12.39</td>
<td>17.43</td>
<td>70.98</td>
<td>72.48</td>
<td>75.26</td>
<td>70.85</td>
<td>21.28</td>
<td>28.36</td>
</tr>
<tr>
<td>&gt;0.1</td>
<td>12.37</td>
<td>17.43</td>
<td>70.97</td>
<td>72.48</td>
<td>75.25</td>
<td>70.85</td>
<td>21.26</td>
<td>28.36</td>
</tr>
<tr>
<td>&gt;0.2</td>
<td>17.21</td>
<td>23.64</td>
<td>71.22</td>
<td>72.67</td>
<td>74.60</td>
<td>70.52</td>
<td>27.98</td>
<td>35.41</td>
</tr>
<tr>
<td>&gt;0.3</td>
<td>24.74</td>
<td>26.91</td>
<td>71.83</td>
<td>72.95</td>
<td>73.80</td>
<td>70.16</td>
<td>37.08</td>
<td>38.89</td>
</tr>
<tr>
<td>&gt;0.4</td>
<td>28.02</td>
<td>29.65</td>
<td>78.18</td>
<td>81.53</td>
<td>65.81</td>
<td>60.27</td>
<td>39.30</td>
<td>39.75</td>
</tr>
<tr>
<td>&gt;0.5</td>
<td>27.96</td>
<td>29.58</td>
<td>78.40</td>
<td>81.70</td>
<td>65.55</td>
<td>60.02</td>
<td>39.20</td>
<td>39.63</td>
</tr>
<tr>
<td>&gt;0.6</td>
<td>66.46</td>
<td>71.53</td>
<td>90.65</td>
<td>90.33</td>
<td>41.44</td>
<td>40.76</td>
<td>51.05</td>
<td>51.04</td>
</tr>
<tr>
<td>&gt;0.7</td>
<td>74.41</td>
<td>80.03</td>
<td>91.15</td>
<td>90.65</td>
<td>37.46</td>
<td>37.12</td>
<td>49.82</td>
<td>50.72</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>79.45</td>
<td>80.98</td>
<td>90.66</td>
<td>90.66</td>
<td>36.64</td>
<td>36.62</td>
<td>50.14</td>
<td>50.43</td>
</tr>
<tr>
<td>&gt;0.9</td>
<td>79.47</td>
<td>81.00</td>
<td>90.69</td>
<td>90.67</td>
<td>36.65</td>
<td>36.62</td>
<td>50.15</td>
<td>50.45</td>
</tr>
</tbody>
</table>
Table 3.3 shows the testing results of VSM model with and without WordNet features. Using WordNet on VSM precision and mean average precision basically experiences a steady rise, the ranges are 21.44% to 87.93% and 69.38% to 87.83% at the threshold value 0 to 0.9, recall value linearly descending from 67.30% to 8.20%. The highest f-measure 46.25% are obtained at the threshold is 0.4. In without WordNet, the assessments of precision 23.13% to 85.71%, mean average precision 67.66% to 87.17%, recall 66.31% to 10.83% and f-measure 34.30% to 45.30% at the threshold value 0 to 0.9. some exceptions occur, it shrinks the performance on precision compare with WordNet on at the threshold values 0, 0.1; mean average precision 0.7, 0.8; recall 0.1 to 0.4 and 0.6 to 0.9; f-measure 0, 0.1, 0.3, 0.6 to 0.9.

Table 3.3 Testing results of VSM model

<table>
<thead>
<tr>
<th>Optimal Threshold Value</th>
<th>P % W</th>
<th>P % W</th>
<th>MAP % W*</th>
<th>MAP % W</th>
<th>R % W*</th>
<th>R % W</th>
<th>F-M % W</th>
<th>F-M % W</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>21.44</td>
<td>23.13</td>
<td>69.38</td>
<td>67.66</td>
<td>67.30</td>
<td>66.31</td>
<td>32.50</td>
<td>34.30</td>
</tr>
<tr>
<td>&gt;0.1</td>
<td>23.27</td>
<td>23.90</td>
<td>70.66</td>
<td>68.44</td>
<td>64.75</td>
<td>65.21</td>
<td>34.23</td>
<td>34.97</td>
</tr>
<tr>
<td>&gt;0.2</td>
<td>28.94</td>
<td>28.33</td>
<td>73.94</td>
<td>71.41</td>
<td>58.30</td>
<td>58.54</td>
<td>38.68</td>
<td>38.18</td>
</tr>
<tr>
<td>&gt;0.3</td>
<td>40.20</td>
<td>37.37</td>
<td>76.77</td>
<td>74.67</td>
<td>50.01</td>
<td>51.02</td>
<td>42.72</td>
<td>43.14</td>
</tr>
<tr>
<td>&gt;0.4</td>
<td>51.46</td>
<td>48.38</td>
<td>81.84</td>
<td>78.78</td>
<td>41.99</td>
<td>42.59</td>
<td>46.25</td>
<td>45.30</td>
</tr>
<tr>
<td>&gt;0.5</td>
<td>68.55</td>
<td>63.93</td>
<td>87.00</td>
<td>86.18</td>
<td>33.05</td>
<td>32.74</td>
<td>44.59</td>
<td>43.30</td>
</tr>
<tr>
<td>&gt;0.6</td>
<td>79.75</td>
<td>76.65</td>
<td>87.95</td>
<td>87.07</td>
<td>26.48</td>
<td>27.01</td>
<td>39.75</td>
<td>40.00</td>
</tr>
<tr>
<td>&gt;0.7</td>
<td>76.95</td>
<td>76.31</td>
<td>79.24</td>
<td>81.90</td>
<td>13.39</td>
<td>17.62</td>
<td>22.82</td>
<td>28.63</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>87.93</td>
<td>84.01</td>
<td>87.83</td>
<td>89.34</td>
<td>8.20</td>
<td>15.31</td>
<td>14.99</td>
<td>25.91</td>
</tr>
<tr>
<td>&gt;0.9</td>
<td>87.93</td>
<td>85.71</td>
<td>87.83</td>
<td>87.17</td>
<td>8.20</td>
<td>10.83</td>
<td>14.99</td>
<td>18.65</td>
</tr>
</tbody>
</table>
Table 3.4 gives the testing results of LSI model with and without WordNet features. Using WordNet, the precision ranges from 4.10% to 76.89%, mean average precision ranges 59.35% to 87.22%; recall ranges from 81.74% to 23.95% and F-Measure ranges from 7.80% to 36.54% at the threshold value 0 to 0.9, and without WordNet, the values of precision 4.7% to 73.06%, mean average precision 58.56% to 83.97%, recall 80.68% to 24.97% and f-measure 7.85% to 42.87% at the threshold value 0 to 0.9. Some exceptions occur it may shrink the performance on precision compare with WordNet on precision at 0.1, 0.2, and 0.3; recall 0.3 to 0.9; f-measure 0, 0.1, 0.2, 0.3, 0.4, 0.8 and 0.9.

<table>
<thead>
<tr>
<th>Optimal Threshold Value</th>
<th>P % W*</th>
<th>P % W</th>
<th>MAP % W*</th>
<th>MAP % W</th>
<th>R % W*</th>
<th>R % W</th>
<th>F-M % W*</th>
<th>F-M % W</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0</td>
<td>4.10</td>
<td>4.7</td>
<td>59.35</td>
<td>58.56</td>
<td>81.74</td>
<td>80.68</td>
<td>7.80</td>
<td>7.85</td>
</tr>
<tr>
<td>&gt;0.1</td>
<td>20.23</td>
<td>21.08</td>
<td>65.94</td>
<td>62.86</td>
<td>70.10</td>
<td>69.49</td>
<td>31.40</td>
<td>32.35</td>
</tr>
<tr>
<td>&gt;0.2</td>
<td>27.18</td>
<td>27.96</td>
<td>69.90</td>
<td>68.80</td>
<td>63.31</td>
<td>62.11</td>
<td>38.03</td>
<td>38.57</td>
</tr>
<tr>
<td>&gt;0.3</td>
<td>32.13</td>
<td>32.82</td>
<td>74.05</td>
<td>72.34</td>
<td>56.61</td>
<td>56.73</td>
<td>41.00</td>
<td>41.58</td>
</tr>
<tr>
<td>&gt;0.4</td>
<td>37.01</td>
<td>36.15</td>
<td>77.57</td>
<td>75.23</td>
<td>50.50</td>
<td>51.51</td>
<td>42.72</td>
<td>42.49</td>
</tr>
<tr>
<td>&gt;0.5</td>
<td>42.39</td>
<td>40.10</td>
<td>82.10</td>
<td>79.33</td>
<td>44.01</td>
<td>45.38</td>
<td>43.19</td>
<td>42.58</td>
</tr>
<tr>
<td>&gt;0.6</td>
<td>51.60</td>
<td>46.76</td>
<td>83.26</td>
<td>80.44</td>
<td>38.21</td>
<td>39.58</td>
<td>43.92</td>
<td>42.87</td>
</tr>
<tr>
<td>&gt;0.7</td>
<td>59.98</td>
<td>55.19</td>
<td>85.11</td>
<td>81.82</td>
<td>31.55</td>
<td>31.87</td>
<td>41.54</td>
<td>40.40</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>76.90</td>
<td>59.65</td>
<td>87.23</td>
<td>82.08</td>
<td>23.97</td>
<td>28.36</td>
<td>36.55</td>
<td>38.50</td>
</tr>
<tr>
<td>&gt;0.9</td>
<td>76.89</td>
<td>73.06</td>
<td>87.22</td>
<td>83.97</td>
<td>23.95</td>
<td>24.97</td>
<td>36.54</td>
<td>37.22</td>
</tr>
</tbody>
</table>

**Graph Results:** The following graphs reveal comparing results of with and without WordNet features and performances of precision, mean average precision, recall and f-measure.
Figure 3.7 displays the comparison of four models on precision with WordNet. The highest precision values of SCBR 98.52%, ECBR 79.47% VSM 87.93% and LSI 76.89%. The four models compared as SCBR>VSM>ECBR>LSI. From the precision graph results the SCBR model 10.59% is higher than second category of VSM model, 19.05% is higher than ECBR model and 21.63% is higher than LSI model, which depicts the SCBR model is able to retrieve precise metadata with concepts.

Figure 3.7 Precision with WordNet
Figure 3.8 shows the comparison of four models on precision without WordNet. The performance of ECBR and SCBR models on this parameter are almost parallel without WordNet. The highest value of SCBR 80.98%, ECBR 81%, VSM 85.71% and LSI 73.06%. The only marginal difference between SCBR and ECBR model without WordNet is 0.02%. It is seen that the SCBR model performs same as ECBR performance in without WordNet. The VSM model is leading in without WordNet. The model compared as VSM>ECBR>SCBR>LSI.

The comparison of SCBR and ECBR are about the same precision values. In comparison, if by using NLP technique such as a spell checker and WordNet (synonymer), gain of high level precision rate is achieved. In contrast, if the NLP techniques are not adopted, the value of SCBR is the same as ECBR. Hence, the proposed applications of NLP techniques will improve the accuracy of SCBR over ECBR and other models.

![Figure 3.8 Precision without WordNet](image-url)
Figure 3.9 displays the comparison of four models on mean average precision with WordNet on. The highest value of SCBR gains of 99.53% when compared with ECBR at 91.15%, VSM at 87.95%, LSI at 87.23%. The models compared SCBR>ECBR>VSM>LSI. Hence, It is observed that the SCBR score is higher than other three IR models.

Figure 3.9 Mean average precision with WordNet
Figure 3.10 displays the comparison of four models on mean average precision without WordNet. The highest score of SCBR is 90.69% as compared with ECBR at 90.67%, VSM at 87.17% and LSI at 83.97%. The models perform as SCBR>ECBR>VSM>LSI.
Figure 3.11 displays the comparison of four models on recall with WordNet. The highest values of SCBR is 80.50%, ECBR is 75.26%, VSM is 67.30% and LSI is 81.74%. These models compared of performance as highest score LSI>SCBR>ECBR>VSM. The LSI model recall ranges higher than SCBR model, 1.24% of the threshold value 0 only. On the contrary, at the threshold value 0.1 to 0.9 the SCBR model gives a better recall rate than the LSI model. The SCBR model, 5.54% is higher when with compared ECBR.

![Figure 3.11 Recall with WordNet](image)
Figure 3.12 displays the comparison of four models on recall without WordNet features. The highest score of SCBR 70.85%, ECBR 70.85%, VSM 66.31% and LSI 80.68%. These models perform as LSI>SCBR=ECBR>VSM.

The comparison of SCBR and ECBR are of the same recall values. In comparison, if by using NLP technique such as a spell checker and WordNet, gain of high level recall rate is achieved. In contrast, if the NLP techniques are not adopted, the value of SCBR is the same as ECBR. Hence, the proposed applications of NLP techniques will improve the accuracy of SCBR over ECBR model.

![Figure 3.12 Recall without WordNet](image-url)
Figure 3.13 displays comparison of four models on f-measure with WordNet features. The highest score is recorded at SCBR 67.56% in comparison with ECBR at 51.05%, VSM at 46.25%, and LSI at 43.92%. Hence, the performance at SCBR is superior to other models. These models compare the performance as SCBR>ECBR>VSM>LSI.

![Figure 3.13 F-measure with WordNet](image_url)
Figure 3.14 displays a comparison of four models on f-measure without WordNet features. The highest values of SCBR 51.93% at the threshold value 0.6, ECBR 51.04% of the threshold value 0.6, VSM 45.30% of the threshold value 0.4, and LSI 42.87% of the threshold value 0.6. Though the performance of SCBR and ECBR are of marginal difference and it is positively superior to LSI and VSM.

**Figure 3.14 F-Measure without WordNet**

**Discussion:** In this section of experimental discussion, comparing the performance of four models SCBR, ECBR, VSM, LSI uses educational dataset based on the four parameters viz, precision, mean average precision, recall and f-measure, the SCBR is stands superior performances. In this experiment set optimal threshold value 0 to 1 with an increment of 0.1 for each model with same set of queries. Table 3.1, 3.2, 3.3 and 3.4 presented testing results of four models in order to SCBR, ECBR, VSM and LSI. Moreover, Figure 3.7, 3.9, 3.11, 3.13 shows graphs results of in order to precision, mean average precision, recall and f-measure using WordNet such
as NLP techniques. Figure 3.8, 3.10, 3.12 and 3.14 shows the graph results of in order to precision, mean average precision, recall and f-measure without using WordNet such as NLP techniques. Here the proposed work compares the performance in two ways. First comparison with four IR model and second with and without using WordNet (NLP techniques such as spell checker and synonymer).

Figure 3.7 displays the comparison of four models on precision with WordNet. The comparison of four models performed as SCBR>VSM>ECBR>LSI. The SCBR model, 10.59% higher than second category of VSM model, 19.05% higher than the third category of ECBR model and 21.63% higher than LSI model. Figure 3.8 shows the comparison of four models on precision without WordNet. The performance of ECBR and SCBR models on this parameter are almost parallel without WordNet. These performed as VSM>ECBR>SCBR>LSI. The level placed in VSM model and the differences between SCBR and ECBR model without WordNet only 0.02%. It is seen that the SCBR model is marginally higher than ECBR performance in without WordNet.

Figure 3.9 displays the comparison of four models on mean average precision with WordNet. These models worked as SCBR>ECBR>VSM>LSI. The SCBR model achieved 8.58% more than ECBR model, 11.58% than VSM model and 12.38% than the LSI model. Figure 3.10 displays the comparison of four models on mean average precision without WordNet. The models perform as SCBR>ECBR>VSM>LSI. Here the SCBR and ECBR performance is only marginally difference by 0.02%. Even though 3.52% higher than VSM model and 6.72% higher than LSI model.

Figure 3.11 displays the comparison of four models on recall with WordNet. These models compare the performance based on highest score
LSI>SCBR>ECBR>VSM. The LSI model recall ranges higher than SCBR model, 1.24% at the threshold value 0. On the contrary, at the threshold value 0.1 to 0.9 the SCBR model gives a better recall rate than the LSI model. The SCBR model of 5.54% higher when compared with ECBR. Figure 3.12 displays the comparison of four models on recall without WordNet. These models perform as LSI>SCBR=ECBR>VSM. The comparison of SCBR and ECBR are of the same recall values. From this comparison, if by using NLP technique such as a spell checker and WordNet, it gains a high level of recall rate, otherwise the rate is same as ECBR model. The high level recall rate means a high level of accuracy results. Hence the proposed model if by applying NLP techniques will improve the accuracy than ECBR model.

Figure 3.13 displays a comparison of four models on f-measure with WordNet. These models compare the performance as SCBR>ECBR>VSM>LSI. Figure 3.14 displays a comparison of four models to f-measure without WordNet. The models perform as SCBR and ECBR in parallel position and next to LSI and VSM.

The above discussion established that SCBR model performs excellently. SCBR model achieved higher results in precision of 98.52%, mean average precision of 99.53% recall of 80.50% and f-measure of 67.56%. From the experimental results the f-measure value is relatively low, therefore, to enhance the f-measure score in chapter 5. As a result SCBR model is able to retrieve precise data with most relevant concepts and the SCBR score is higher than other three models based on precision, mean average precision, recall and f-measure. It is this reason the NLP techniques are adopted. The spellchecker and WordNet are responsible for to have high recall at the expensive or precision of SCBR model. In conclusion, the performance of SCBR model stands superior to the three other IR models.
3.5 SUMMARY OF THE CHAPTER

The chapter proposed a reliable and an efficient generic semantic search technique system, which suggests the user all the effective details to know about an educational domain. It is reliable because though it is being inputted with synonymous words and misspell, it retrieves the similar result and does not provide an irrelevant result. The spell checker advantage to user queries can improve the effectiveness of search and efficiency of obtaining useful data. In this research, the most efficient SCBR algorithm is designed. The SCBR is derived from ECBR algorithm, in structure to compute the similarity values between query and concepts. This chapter compares the performance of the SCBR model with three information retrieval models. To address the defect of low recall rate that is done in ECBR model. The research modified ECBR algorithm to SCBR algorithm obtains better performance, particularly precision and recall. Moreover, in this chapter, NLP techniques like spellchecker and synonymer is used for proficient meaningful search. In the next chapter propose a specific semantic search technique for with domain knowledge user.