CHAPTER 6

SEMANTIC MATCHING APPROACH WITH IR

6.1 INTRODUCTION

The traditional information retrieval technique performs keyword searching that matches with the keywords specified in their queries. The well known example is Google, where entered keywords may not be taken for semantic matching in order to retrieve from the web. Synonym and polysemy are two important issues to be resolved when IR should support as semantic matching. A synonym is a word that means the same as another word and a polysemy is a word with multiple related meaning. In IR based on semantic should have higher precision than the other system [49].

Semantic matching plays an important role in information retrieval (IR). IR should give results with respective to same meaning even if the entered string is not exact keyword. In multilingual environment, a user fires a query in one language and retrieves the information either in the same language or in different language after semantically matched. This is highly needed in rural areas, where most of the time information provided and acquired depends on semantic matching. There are numerous approaches proposed to match semantically such as first, by applying translation of the keyword into target language using machine translation or using dictionary. Second, use of ontology to represent the knowledge of a domain and match that ontology for semantic.

The main problem arise with these approaches are how to build ontology, how to traverse it and acquire the knowledge from it to make an inference. The performance of a system is highly improved when we use ontology and its structure to represent the knowledge.

In this chapter, we focused on proposed ontology building approach from database which is used for semantic matching. We provide user friendly environment as a system, which will ask or provide certain questions based on sub-domain or domain. From this, the user will be acquainted about certain things of the sub-domain or he may provide his own knowledge to build ontology. This approach has advantages that ontology will be built from exact terms
with the relationships among them with, no redundancy in knowledge representation. Finally ontology has been shown in hierarchical form.

The main objective is to make an inference from ontology as a result of a query. We extended above approach as an inference approach which is called as IR, where the inference can be calculated from ontology irrespective of language or matching. The approach is supported by synset for each language instead of WordNet, since the approach is for a particular domain.

6.2 SEMANTIC MATCHING METHODOLOGY

Once ontology has been built from database for a sub-domain by using above proposed approach, the entire system can be extended as an inference system. Figure 6.1 shows the overall system architecture for semantic-based IR using ontology. The system includes five modules namely user interface module, query formation module, ontology module, inference engine module and IR with display module. Each module has been briefly described below.

6.3 IMPLEMENTATION DETAILS

The semantic matching approach used external resources such as bi-lingual dictionaries, synsets and so on. Using these resources, each query word can be searched either in dictionary or in synset in order to extract the equivalent keyword. We used ontology approach in order to fast and accurate results of semantic matching as IR. The design of an approach is elaborated in section 6.4 and the matching algorithm for IR is given in section 6.5. The results are being used as inference results after semantic matching.

6.4 DESIGN

Each component of the system architecture is explained in brief.

- User interface with local language support:

  This module gives the user interface in order to input the parameters required for an inference query for the sub-domain. This is supported in local languages such as
Hindi or Marathi. Either user selects single or compound keywords from user interface to make an inference from the system. We made a simple user interface supported by choice of ontology terms for the user.

Query Formation Module:

The selected keyword in local language is being translated into English by using bi-lingual dictionary and formed SQL query to search into ontology.

Ontology Module:

This module provides a knowledge pool in the form of ontology. We are representing ontology in a hierarchical form with sub-concepts being related by some relations such as IS-A, Has etc. The entire ontology will be traversed to match as concept, sub-concept, property or property’s value for a query keyword. If found in ontology then extract those ontology terms and make inference from them.
Inference Engine:

Once all the terms from ontology have been found, they are given to this module. According to the occurrence of the keyword/s entered in a query, its places and the relations between them the ontology can be traversed forward or backward. The matching terms can be found out. An inference can be determined easily with these terminologies from ontology. External resources have been used as bi-lingual domain dictionary for translation and synsets to extract corresponding keyword when they are not found in domain dictionary for each language.

IR and Display in Local Language:

Finally the result of inference engine is forwarded to this module for display purpose. This module handles two functionalities, first, it handles the information retrieval as knowledge from the ontology, and second, it displays the inference result in proper format in selected local language.

Implementation Strategy

Our strategy for inference includes a user interface to enter a query in local languages such as Hindi or Marathi. Then we translate or transliterate the query terms into English by using bi-lingual domain dictionary and extracted the ontology terms from it. These terms are searched in the ontology from top to bottom to get exact terms and their relationships. If the terms are not found in ontology then alternative keywords are being searched bi-lingual domain specific synsets. Finally, an inference engine will compute the inference based on relationships and terms from ontology.

6.4.1 ONTOLOGY BUILDING METHODOLOGY

6.4.1.1 Need of Ontology for Semantic Matching

Let us consider a sub-domain as ‘puri-chhole’ or ‘ragda-patis’ from a domain ‘Grocery shop’ as case study. User either comes to the shop or directly access the system kept in a
shop to acquire the knowledge or access the system online to get the necessary service. According to the need of user, a scenario can be generated in the form of questions-answers that leads to a knowledge pool about that domain or sub-domain. System provides the necessary user interfaces to proceed with the sub-domain to acquire the knowledge. Let us assume that the user does not know much about the items needed for the menu and the procedure to make. The system asks some questions and gets the answers from the user in his local language. Those answers can be used to build ontology which can be used to represent the knowledge about the menu.

If a system doesn’t interpret the keyword entered by the user then the desired information may not be acquired by the user. In this scenario, if the entered keyword matches semantically then the system can provide the required information to the user. This information can be accurate and complete when it is represented in an efficient way. Ontology gives the exact and efficient environment to represent the knowledge.

6.4.1.2 **Knowledge Pool for the user**

Once the user provides the answers in his local language, those are converted into English and checked or searched with the database for providing necessary knowledge. According to the proposed algorithm ontology can be built. This ontology is used as a knowledge pool for the sub-domain. The user can be prompted to make certain inference about the menu, which may take help of ontology. The knowledge can be acquired from the ontology knowledge pool. Different ontology can be build dynamically for each user scenario.

6.4.2 **IMPLEMENTATION DETAILS OF ONTOLOGY BUILDING METHODOLOGY**

6.4.2.1 **Design**

The overall system architecture of building ontology from database is shown in figure 6.2. The entered local language keyword is being translated or transliterated into English by translation or transliteration module. We use the machine-readable bi-lingual domain-based Hindi-> English and Marathi->English dictionaries for keyword translation. In this dictionary, we maintained the entries which are common to the items in all these three
languages. But in literature [37] it was suggested that the bilingual dictionaries may contain more verbose definitions which are not suitable for retrieval. We had correct and accurate entries since we are using domain-based entries in the dictionary. Hindi and Marathi are morphologically rich [1, 52]. Therefore, their keywords are stemmed before looking up in the bi-lingual dictionary. If the word is found in a dictionary then it goes to the database to find exact match for the ontology. In case a match is not found, we transliterate to get exact match from the dictionary and searched again in database. Ontology is built from the database keywords.

Figure 6.2 shows the proposed overall system architecture for ontology building for semantic matching approach. The system includes seven modules as user interface module, parsing module, Q-A module, stemmer module, Translation or transliteration module, query module and ontology module. These modules are described as follows.

- **User Interface Module:** This module is a user interface module which provides the interactive environment for the user to use the system. The user is prompted to select the choice of his or her local language along with sub-domain. The sub-domain
keyword may be simple or complex. Further those strings are parsed by the parsing module.

- **Parsing Module:** The entered keyword is interpreted by this module as complex or simple. The two keywords combining represent the sub-domain. This module takes two strings entered as input or selected from the menu as input; parse them to form two separate keywords. Question-Answer (Q-A) module is prompted for each keyword.

- **Question-Answer (Q-A) Module:** This module is an important module since it leads to the actual building of the ontology. Our system provides set of questions according to the entered keyword. These questions has been developed in such a fashion that the user is able to know the complete coverage of sub-domain. The probable answers for these questions are provided as drop-down menu or user may enter the expected answer according to his or her knowledge. This module takes care of local language input for the question-answer set, which is forwarded to the translation module.

- **Stemmer Module:** This module takes the input as answer from question-answer module and stems the keyword by removing all stop words to get the root word. This root word is given to translation module for translation into English.

- **Translator or Transliteration Module:** Once the entered local language keyword is interpreted as answer for the prescribed question, this module provides the translation from one language to English language by using bi-lingual dictionary. This module directly translates the entered answer or transliterates the answer in order to get its equivalent English keyword. Once the Hindi or Marathi keyword is translated into English, it is passed to the query module to generate a query.

- **Query Module:** The keyword from translation or transliteration module being taken to form SQL query according to question-answer module. The answer of each question from Q-A module takes to form a query. Query is passed to the database for searching and result is being accessed.

- **Ontology Building Module:** This module actually builds ontology by accessing one-by-one keyword from database after matching with answers from user. According to the proposed algorithm, each keyword is placed to form hierarchy to represent
ontology as per rules defined for ontology. Only, we have to represent the relationships among the concepts and sub-concepts explicitly.

6.4.3 IMPLEMENTATION STRATEGY

We have implemented the approach according to the proposed algorithm 6.3.3 for building ontology from database. Some rules have been developed for construction of ontology from database. With these rules we can represent database schemas as ontology by mapping the terms from both [44]. Some rules are given below.

6.4.3.1 Translation or Transliteration Rules

Rule 1: Entered keyword is transliterated into English after parsing the keyword into vowel, consonant or modifier by transliteration module.

Rule 2: Acquire its meaning from the domain-based dictionary. If it does not have any meaning then its transliteration is considered as it is else follow rule 3.

Rule 3: It is translated into English by using bi-lingual domain-based Hindi-English and Marathi-English dictionary.

6.4.3.2 Query formation Rules

Rule 4: Every translated keyword forms SQL query from question-answer module.

6.4.3.3 Transformation from relational database schema information to ontology

Rule 5: If the entered keyword is a table or relation name then it maps as a class for concept.

Rule 6: If the relation R does not contain foreign key then R becomes concept.

Rule 7: For relations R1 and R2, if R2 contains a foreign key as R1 primary key then R1 becomes concept and R2 will be sub-concept.

Rule 8: If the entered keyword is an attribute name of certain relation R then its relation name will be sub-concept or concept depending on presence of foreign key.

6.4.3.4 Transformation from relational database attributes value to ontological instances
Applying above rules, ontological structure has been constructed from extracted database. We can apply following rules to form leaf node as instance values for sub-concept.

Rule 9: If the entered keyword is an attribute value then retrieve the relation with name, attribute name and its value from database. Depending on rule 4, ontology is being formed and represented in hierarchical form.

Rule 10: Extracted value is leaf node for ontology.

Figure 6.3 shoes the mapping of relation schema and ontology by using translation rules.

6.4.4 ONTOLOGY BUILDING: THE ALGORITHM

Objective: Ontology Building from database
Input: Hindi or Marathi keyword/s to represent ontology for sub-domain
Output: Ontology for a sub-domain

1. Enter the keyword to acquire knowledge from the database.
2. If the keyword is simple, go to step 4 else go to step 3.
3. Parsing module: Parse the complex keyword to get proper keywords then goes to step 4.
4. Translate or transliterate the keyword by using domain-based dictionary.
5. Question-Answer (Q-A) module: The user asked to answer the questions by using Q-A interface for selected sub-domain.
6. Each answer is being translated or transliterated into English by using dictionary and SQL query is formed.
7. Ontology building module: Search for table name or attribute name or attribute value in the database for the entered keyword.
8. Map the retrieved data to ontology terms according to translation rules.
9. Form the hierarchy of ontology terms from the data to display ontology according to sequence of rules.
10. Define the relationships as ‘is-a’ and ‘has’ within the concepts and sub-concepts of ontology.

EXAMPLE

Consider a sub-domain as ‘पूरी छोले’ from ‘Grocery shop’ domain. Let us find out possible ontology terms which may be useful to the user. The complex keyword ‘पूरी छोले’ is parsed to get two keywords as ‘पूरी’ and ‘छोले’. For each keyword, we has separate question-answer module to get ontology terms from database. For example, for ‘पूरी’, the question may be ‘क्या आप पूरी के लिए गेहूँ या गेहूँका आटा पसंद करोगे?’ and possible answer may be ‘गेहूँ या गेहूँका आटा’ or user is prompted to enter the possible answer. With this answer, the selected or entered keywords are stemmed as ‘गेहूँ’ and ‘आटा’; both keywords are translated into English and searched in the database to get concept, sub-concept and relationships for ontology.

This way each answer of question from Q-A module is translated into English and formed SQL query to get answers from database for ontology.
Ontology Representation in English (Hierarchical form)

Figure 6.4 shows the possible ontology for ‘पूरी छोले’ with hierarchy of concepts, sub-concepts and attributes. The possible relationships are considered such as ‘needs’, ‘used-for’, ‘is-a’ or ‘has’, for example, ‘is-a’ is a relationship between wheat and complete-grain or type & price are the attributes of wheat. Table 6.1 shows the representation of ontology with relationships; we can calculate the ancestor or successor of any node within ontology. This helps for inference analysis.

**Puri-Chhole**

- **Puri** (Concept)
  - **Grain** (Concept)
    - **Complete-grain** (Sub-concept)
    - **Wheat** (Attribute)
      - **Wheat** (type, price) (Values)

Figure 6.4: Ontology for ‘पूरी छोले’

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Ancestor</th>
<th>Successor</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wheat (leaf-node)</td>
<td>Complete-grain</td>
<td>Type, availability, price (attributes)</td>
<td>has</td>
</tr>
<tr>
<td>Complete-grain</td>
<td>Grain</td>
<td>Wheat</td>
<td>has/is-a</td>
</tr>
<tr>
<td>Grain</td>
<td>Puri</td>
<td>Complete-grain</td>
<td>has/is-a</td>
</tr>
<tr>
<td>Puri</td>
<td>NIL</td>
<td>Grain</td>
<td>needs/used-for</td>
</tr>
</tbody>
</table>

Table 6.1: Representation of Ontology with relationships
6.4.5 EXPERIMENTAL SETUP

6.4.5.1 Experimental Requirements

The requirements for the research work are as follows:

6.4.5.1.1 Technical Requirements

1. Hardware: Pentium IV PC with minimum 1GB RAM, 900MHz processor, minimum 60GB HD.

2. Software: Core JAVA for user interfaces, database system to handle data such as MS SQL Server 2005, IMEs for Hindi and Marathi languages for input local language strings, Notepad ++ editor for java programs.

6.4.5.1.2 Non-Technical Requirements

1. Number of users as customers: 5
2. Number of Domain: 1 (Grocery shop)
3. Number of Sub-domains: 2 (Puri-Chhole and Ragda-Patis)
4. Number of external resources: 1 (domain database)

6.4.5.2 Performance Parameters for Ontology Building

There are many performance parameters to evaluate the approach. Some are considered for this research work as follows.

1. Number of users as customers

There are 5 users as customers to use system for a particular sub-domain. Each user uses the same sub-domain, but their answers for questions can be different. All are looking for knowledge from database where ontology is being built.
2. Total ontology terms per user

This gives the exact number of terms extracted from the database to build ontology for each user. For each user, the number can be different even if they use the same sub-domain. These terms depend on the context and answers to the questions.

3. Time taken to build ontology for each user

The time required to build ontology for a user in milliseconds. This also measures the accuracy as well as speed of a system supported by the machine.

4. Accuracy of building the ontology

This defines the number of users entering the correct input as answers which are easily used by the system to build ontology from database. It is the degree of closeness of measurements of a quantity to that quantity’s actual value. It is defined as -

\[
\text{Accuracy} = \frac{((A) + (B))}{(A + B + C + D)}
\]

Where –

\[A – \text{Number of true positive}\]
\[B – \text{Number of true negative}\]
\[C – \text{Number of false positive}\]
\[D - \text{Number of false negative}\]

4.5 Precision and Recall

These parameters define how accurately extraction and use of terms to build ontology from database. The precision is defined as -

\[\text{Precision} = \frac{A}{(A+C)} \times 100 \text{ in percentage.}\]

And recall is defined as –

\[\text{Recall} = \frac{A}{(A+B)} \times 100 \text{ in percentage.}\]
Where -

A – Number of relevant terms retrieved to build ontology from database.
B – Number of relevant terms but not retrieved to build ontology from database.
C – Number of irrelevant terms from database to build ontology from database.

5. F-measure

F-measure can be measured from precision and recall values from ontology. It is defined as harmonic mean of precision and recall as –

\[
F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

6.4.6 EXPERIMENTAL RESULTS

We evaluated our system in various aspects. First, we compared the ontology terms needed to build ontology for five users, second; we compared the time taken by each user in ms to build ontology. Again the time depends on many factors like CPU processing speed, hardware support, size of RAM etc. We evaluated our system by calculating three important parameters like accuracy, precision and recall. Table 6.2 shows total ontology terms and time in ms required to build ontology for a sub-domain ‘puri-chhole’ for Hindi and Marathi users. Table 6.3 shows total ontology terms and time in ms required to build ontology for a sub-domain ‘ragda-patis’ for Hindi and Marathi users. Table 6.4 shows the performance parameters for Hindi and Marathi users for both the domains.

6.4.6.1 ‘Puri-chhole’ sub-domain Results

Table 6.2: Total Ontology Terms and Time in ms to build Ontology for Hindi and Marathi Users

<table>
<thead>
<tr>
<th>Sr. no.</th>
<th>Users</th>
<th>Total Ontology Terms</th>
<th>Time to build ontology in ms</th>
<th>Total Ontology Terms</th>
<th>Time to build ontology in ms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Hindi user</td>
<td></td>
<td>Marathi user</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>User 1</td>
<td>258</td>
<td>234</td>
<td>258</td>
<td>203</td>
</tr>
<tr>
<td>2</td>
<td>User 2</td>
<td>258</td>
<td>172</td>
<td>258</td>
<td>156</td>
</tr>
<tr>
<td>3</td>
<td>User 3</td>
<td>258</td>
<td>203</td>
<td>258</td>
<td>188</td>
</tr>
<tr>
<td>4</td>
<td>User 4</td>
<td>258</td>
<td>173</td>
<td>258</td>
<td>63</td>
</tr>
<tr>
<td>5</td>
<td>User 5</td>
<td>258</td>
<td>203</td>
<td>258</td>
<td>109</td>
</tr>
</tbody>
</table>
Information Retrieval in Multilingual Environment: Phonetic and Semantic Along with Ontology Approaches for Hindi and Marathi

Figure 6.5: Time required to build Ontology for Hindi Users

Figure 6.6: Time required to build Ontology for Marathi Users

Figure 6.7: Comparison of Time in ms required to build ontology for Hindi & Marathi Users
Information Retrieval in Multilingual Environment: Phonetic and Semantic Along with Ontology Approaches for Hindi and Marathi

Figure 6.8: Precision and Recall for Hindi Users for Building Ontology

Figure 6.9: Precision and Recall for Marathi Users for Building Ontology
6.4.6.2 ‘Ragda-patis’ sub-domain Results

Table 6.3: Total Ontology Terms and Time Required to Build Ontology for Hindi and Marathi Users

<table>
<thead>
<tr>
<th>Sr. no.</th>
<th>Users</th>
<th>Total Ontology Terms</th>
<th>Time to build ontology in ms</th>
<th>Total Ontology Terms</th>
<th>Time to build ontology in ms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hindi user</td>
<td>Marathi user</td>
<td></td>
<td>Hindi user</td>
<td>Marathi user</td>
</tr>
<tr>
<td>1</td>
<td>User 1</td>
<td>432</td>
<td>186</td>
<td>432</td>
<td>94</td>
</tr>
<tr>
<td>2</td>
<td>User 2</td>
<td>432</td>
<td>109</td>
<td>432</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>User 3</td>
<td>432</td>
<td>281</td>
<td>432</td>
<td>142</td>
</tr>
<tr>
<td>4</td>
<td>User 4</td>
<td>432</td>
<td>281</td>
<td>432</td>
<td>125</td>
</tr>
<tr>
<td>5</td>
<td>User 5</td>
<td>432</td>
<td>140</td>
<td>432</td>
<td>140</td>
</tr>
</tbody>
</table>

Figure 6.10: Time in ms required to build Ontology for Hindi Users

Figure 6.11: Time in ms required to build Ontology for Marathi Users
Figure 6.12: Comparison of Time in ms required to build ontology for Hindi & Marathi Users

Figure 6.13: Precision and Recall for Hindi Users for Building Ontology

Figure 6.14: Precision and Recall for Marathi Users for Building Ontology
Table 6.4: Accuracy, Precision and Recall for Ontology Building

<table>
<thead>
<tr>
<th>Sub-Domain</th>
<th>Users</th>
<th>Hindi</th>
<th></th>
<th></th>
<th></th>
<th>Marathi</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A</td>
<td>P</td>
<td>R</td>
<td>A</td>
<td>P</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Puri-Chhole</td>
<td>User 1</td>
<td>100.00</td>
<td>100.00</td>
<td>27.13</td>
<td>100.00</td>
<td>100.00</td>
<td>27.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 2</td>
<td>100.00</td>
<td>100.00</td>
<td>20.93</td>
<td>100.00</td>
<td>100.00</td>
<td>20.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 3</td>
<td>100.00</td>
<td>100.00</td>
<td>23.25</td>
<td>100.00</td>
<td>100.00</td>
<td>23.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 4</td>
<td>100.00</td>
<td>100.00</td>
<td>18.21</td>
<td>100.00</td>
<td>100.00</td>
<td>18.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 5</td>
<td>100.00</td>
<td>100.00</td>
<td>27.13</td>
<td>100.00</td>
<td>100.00</td>
<td>27.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ragdapatteis</td>
<td>User 1</td>
<td>83.79</td>
<td>47.10</td>
<td>15.74</td>
<td>83.79</td>
<td>36.93</td>
<td>11.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 2</td>
<td>83.79</td>
<td>44.88</td>
<td>15.74</td>
<td>83.79</td>
<td>41.66</td>
<td>13.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 3</td>
<td>83.79</td>
<td>44.88</td>
<td>15.74</td>
<td>83.79</td>
<td>44.88</td>
<td>15.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 4</td>
<td>83.79</td>
<td>44.88</td>
<td>15.74</td>
<td>83.79</td>
<td>40.67</td>
<td>13.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User 5</td>
<td>83.79</td>
<td>41.66</td>
<td>13.81</td>
<td>83.79</td>
<td>24.73</td>
<td>6.35</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where A: Accuracy in %, P: Precision in %, R: Recall in %.

6.4.7 DISCUSSION

The entire system has two main phases, first, ontology building and second, semantic matching with inference. In the ontology building phase, we proposed an integrated approach to build ontology from database, in which issues like extraction of data from database, form a hierarchy to represent ontology and traversing it forward and backward have been handled efficiently. The entire system has many phases like collection of data for a domain, its representation as knowledge, searching and traversing, retrieving and display in proper format, providing user interfaces to support Hindi and Marathi languages and so on.
6.5 SEMANTIC MATCHING: THE ALGORITHM WITH IR

Objective: Semantic Matching using Ontology
Input: Query terms in Hindi or Marathi
Output: Semantic Matching with inference system as IR

1. Enter or select question/s for the sub-domain to make an inference.
2. Translate or transliterate the query terms into English by using bi-lingual dictionary.
3. Each query term is mapped with ontology terms from ontology.
4. Traverse the ontology forward or backward for each term.
5. If found in ontology then retrieve the ontology terms such as concept, sub-concept, attribute name and/or attribute value and their relationships, go to step 8.
6. If not found in ontology then search in the domain synsets for its alternative keyword.
7. Get the corresponding synset, find its equivalent English synset from bi-lingual synset and search again in ontology. If found then the term can be called as inference result.
8. Acquire the inference by using relationships from the ontology.
9. Finally, translate all the inference strings and display the extracted result from ontology as inference for the query as IR.

EXAMPLE

Consider a query in Hindi as, ‘पुरी के लिए कोनसा गेहुँ चाहिए?’ to be executed by the system for a grocery shop domain. If it is given to any database system then we may not acquire any result. The reasons are first, database system is not maintaining the attribute value as ‘पुरी’ in any language and second this item is not related to grocery shop items. The SQL query may be ‘select wheat_type from grains where item = ‘पुरी’’.

In order to retrieve the information for this query we need another approach. The name of the approach is ‘inference system’ approach. An inference query can be built and fired at
ontology, where we can find out the inference from relationships within the concepts. In this example, the inference answer is the name of the wheat from ontology.

6.6 EXPERIMENTAL SETUP FOR SEMANTIC MATCHING

The increasing number of methods available for semantic matching suggests the need of evaluation of these methods. Similar to existing approaches, we had evaluated the system with different performance parameters as mentioned below. These help the system designers to assess the strengths and weaknesses of their systems as well as help application developers to choose the most appropriate algorithms.

By considering the case study ‘Grocery Shop’ as domain for semantic matching, we evaluated the system by considering the number of users as customers who wish to acquire the information. We evaluate the system by considering the standard performance parameters like precision, recall, F-measure and accuracy.

6.6.1 EXPERIMENTAL REQUIREMENTS

The requirements for the research work are as follows:

6.6.1.1 Technical Requirements

1. Hardware: Pentium IV PC with minimum 1GB RAM, 900MHz processor, minimum 60GB HD.
2. Software: Core JAVA for user interfaces, database system to handle data such as MS SQL Server 2005, IMEs for Hindi and Marathi languages to input the local language strings, Notepad ++ editor for java programs.

6.6.1.2 Non-Technical Requirements

1. Number of users as customers: 2
2. Number of Grocery shop owner: 1
3. Number of Domain: 1 (Grocery shop)
4. Number of Sub-domains: 2 (Puri-chhole and Ragda-Patis)
5. Number of Bi-lingual dictionary: 1 (Hindi-English-Marathi)
6. Number of synset: 1 (Hindi-English-Marathi)

6.6.2 Performance Parameters

There are many performance parameters to evaluate the approach. Some are considered for this research work, are as follows.

1. Number of users as customers

There are two users as customers to use system for a particular sub-domain. We assumed that one user each from Hindi and Marathi linguist community, who will use the system for inference. All are looking for knowledge from database which can be used for inference.

2. Number of Inference queries

We assumed three inference queries each from Hindi and Marathi linguistic users.

3. Precision and Recall

These parameters define how accurately extraction of terms taken from ontology and used for an inference system. The precision is defined as -

$$\text{Precision} = \frac{A}{(A+C)} \times 100 \text{ in percentage.}$$

And recall is defined as –

$$\text{Recall} = \frac{A}{(A+B)} \times 100 \text{ in percentage.}$$

Where -

A – Number of relevant terms retrieved from ontology for inference.
B – Number of relevant terms but not retrieved from ontology for inference.
C – Number of irrelevant terms from database from ontology for inference.
4. **F-measure**

F-measure can be measured from precision and recall values from ontology. It can be defined as harmonic mean of precision and recall as –

\[
F\text{-measure} = \frac{(2 \times \text{precision} \times \text{recall})}{(\text{precision} + \text{recall})}
\]

5. **Accuracy**

This defines the number of users entering the correct input as answers which are easily used by the system to build ontology from database to make inference. This also counts number of terms extracted from ontology for inference. It is the degree of closeness of measurements of a quantity to that quantity’s actual value. The accuracy can be defined as-

\[
\text{Accuracy} = \frac{(A) + (B)}{(A + B + C + D)}
\]

*Where –*

- \(A\) – Number of true positive
- \(B\) – Number of true negative
- \(C\) – Number of false positive
- \(D\) - Number of false negative

### 6.7 STRATEGY FOR SEMANTIC MATCHING WITH INFERENCE

We assumed that both the users may use the same sub-domain. Once the sub-domain has been selected, the system guides for proper approach starting with the questionnaires for each linguist’s user. Table 6.5 shows the three inference queries for each Hindi and Marathi linguistic user for the sub-domain ‘puri-chhole’ and ‘ragda-patis’. Each query has been interpreted and ontology terms have been extracted from them. Both the terms are being matched; results are acquired and displayed from ontology with their relationships.
Table 6.5: Inference Queries for the Sub-Domains for Hindi and Marathi Users

<table>
<thead>
<tr>
<th>Sub-domain</th>
<th>Query No.</th>
<th>Query Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puri-chhole</td>
<td>Q1</td>
<td>पुरी के लिए कौनसा गेहू इस्तेमाल कर सकते हैं ?</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>छोले के लिए कौनसा दाल इस्तेमाल कर सकते हैं ?</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>छोले के लिए कौनसा मसाला इस्तेमाल कर सकते हैं ?</td>
</tr>
<tr>
<td>Ragda-patis</td>
<td>Q1</td>
<td>रगडा साठी कोणता मसाला वापर करू शकतात ?</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>पंटीस साठी कोणते तेलचा वापर करू शकतात ?</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>रगडा साठी आणकी ल काईया वापर करू शकतात ?</td>
</tr>
</tbody>
</table>

6.8 EXPERIMENTAL RESULTS

Table 6.6 shows the precision, recall and F-measure calculated on ontology-based an inference approach. We acquired adequate results for both the sub-domains. Figure 6.15 and figure 6.16 shows the graphs as results for the Hindi and Marathi user in terms of precision, recall, f-measure and accuracy for inference system.

Table 6.6: Precision, Recall, F-measure and Accuracy in % for Hindi and Marathi Users for Inference Approach

<table>
<thead>
<tr>
<th>Sub-domain</th>
<th>Query No.</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puri-chhole</td>
<td>Q1</td>
<td>26.00</td>
<td>70.00</td>
<td>37.91</td>
<td>33.80</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>50.00</td>
<td>100.00</td>
<td>66.66</td>
<td>50.00</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>50.00</td>
<td>100.00</td>
<td>66.66</td>
<td>50.00</td>
</tr>
<tr>
<td>Ragda-patis</td>
<td>Q1</td>
<td>20.00</td>
<td>100.00</td>
<td>33.33</td>
<td>20.00</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>39.00</td>
<td>65.50</td>
<td>48.80</td>
<td>50.00</td>
</tr>
<tr>
<td></td>
<td>Q3</td>
<td>50.00</td>
<td>100.00</td>
<td>66.66</td>
<td>50.00</td>
</tr>
</tbody>
</table>
6.9 DISCUSSION

In this section, we proposed information retrieval after semantic matching using ontology for Hindi and Marathi. Semantic matching is very important in multilingual environment, where many external resources are being used for semantic matching such as dictionaries, synset and so on. We called this IR as a result of inference system after semantic matching. Many
approaches have been proposed for inference system, but use of ontology will optimize the matching procedure also the degree of accuracy of the result may increase.

We evaluated the system with performance parameters such as precision, recall, f-measure and accuracy. By looking at the results of those parameters for our approach, it is found that our approach gives better and accurate results as compared to other approaches.

6.10 SUMMARY

In this chapter, we proposed an approach for semantic matching using ontology as inference system used for IR in multilingual environment. This approach is supported by our proposed integrated ontology building approach, which governs with the rules that built the ontology from database.

First, we proposed an integrated approach for ontology building from database, which gives an easy and simple environment to build ontology. Second, we extended the approach so that multilingual users will use the system effectively for inference. This approach is supported by ontology for semantic matching. After semantic matching has been done, then either Hindi or Marathi user may use the system in order to compute the inference. This inference has been shown as IR for the query. Finally we evaluated both the approaches of inference system for better accuracy.