This paper proposes a method for integrating heterogeneous ontologies using a fuzzy rule-based system and boosting algorithm. The proposed method has a wider scope in knowledge searches, Natural Language Processing, etc. In today's computer world, the Internet plays a main role. Whenever the user searches for information in the search engine, sometimes irrelevant information is retrieved since most of the search engines are based on the Keyword search. Retrieving more relevant and accurate information is a major issue. To enhance the retrieval accuracy, there is a transition from present day web to semantic web. The main application of Information Retrieval is Web Search Engines, where the web is classified as Web 1.0, Web 2.0 and Web 3.0. Web 1.0 is Read Only Web, where users can access and share the documents. Web 2.0 is Read Write Web which is a present day web, users can share photos, videos, interact in social network, publish contents in blog and so on but the limitation is poor in describing the meaning of the content, documents cannot be processed and irrelevant documents are retrieved while searching. To overcome these limitations of Web 2.0, Web 3.0 Semantic Web is introduced it structures the meaningful contents of unstructured documents to make the information machine readable, understandable and processable. Semantic Web processes Complex Queries, Provide Web Services, Data Repositories and agents. The Web 3.0 depends on ontology. Ontology is a structural framework that is a formal way of organizing data. Ontology provides a form of arranging the information into categories and relating these categories with each other. Ontology supports both hierarchical as well as non-hierarchical relations. A hierarchical or taxonomic relation provides both subclass and super class hierarchy. For example, tiger is a subtype of cat. Non-hierarchical relations are based on synonym, antonym and hyponym. For example, Tiger is an animal which is related to Asia the continent to which it belongs. The main reason for the retrieval of irrelevant information is due to the use of polysemy and synonym. This problem can be solved by using ontologies by applying query expansion techniques. Query Expansion techniques are used to find the relatedness between words or concepts semantically by evaluating the user query and intensifying the search query to match the additional document. Semantic relatedness denotes the degrees to which words are associated (synonymy, functional, hypernymy, hyponymy, metonymy associative and other types). To avoid the problem of uncertainty between OWL files and to achieve the greater interoperability between OWL files fuzzy logic is used. For example, “a fun holiday” which is a vague concept, to avoid this sort of uncertainty fuzzy logic is used. The vague concepts can be described through a certain degree of membership function (lies between 0 to 1) which is described by a many-valued logic that is fuzzy logic. To resolve the problem of uncertainty and to retrieve the relevant document fuzzy logic is combined with ontologies. To improve the accuracy of retrieval of relevant information boosting algorithms like AdaBoost and RankBoost Algorithms can be used. This paper is arranged as follows: Section II describes System Model, Section III describes Review of Literature, Section IV shows the implementation part, and Section V depicts the accuracy of the proposed system.

II. SYSTEM MODEL

In the Proposed Approach OWL files are given as the input for query expansion where any number of OWL files of same domain is given. First, the structural similarity between two concepts from two different ontologies is calculated using Wu & Palmer approach [1]. Secondly, the semantic similarity between two concepts from two different ontologies is calculated using Linguistic Measure [6]. Finally the value from both the measures is given to the Fuzzy System. Using Fuzzy system four different rules are formulated as very low, low and high. Depending
upon the rules the vague concepts are classified in any one of the rules. Then the similar concepts are retrieved.

Fig 1: Information Retrieval Methodology
To further optimize the results, boosting algorithms like AdaBoost and RankBoost algorithms are used. Finally, the similar concepts are retrieved and the performance is calculated using Precision and Recall measure for user-defined dataset as well as for benchmark dataset. The proposed method is depicted in the fig 1.

III. IMPLEMENTATION
The query expansion techniques is implemented by the following modules,
- Ontology Creation Using Protégé
- Implementation of Similarity Measures
- Generation of Fuzzy Rules
- Measuring Semantic Relatedness using AdaBoost Algorithm
- Measuring Semantic Relatedness using RankBoost Algorithm

ONTOLOGY CREATION USING PROTÉGÉ
Two different Ontologies that is OWL files are created for the same University domain with different set concepts. These two ontologies are created using the tool called Protégé. Ontology1 consists of 250 concepts and Ontology2 consists of about 270 concepts which are arranged in hierarchical and non-hierarchical manner.

IMPLEMENTATION OF SIMILARITY MEASURES
STRUCTURAL SIMILARITY BETWEEN CONCEPTS (WU & PALMER MEASURE)
Wu and Palmer’s [4] used number of relationships between the concepts to be compared to calculate the similarity. A common ancestor c of c1 and c2 which occurs at the lowest position in the ontology hierarchy (least common super-concept) is considered for calculating the similarity value. Here all the relations between the concepts are assumed to have same value. The similarity is given by:

$$\text{sim}(c1, c2) = \frac{2H(D1 + D2 + 2H)}{D1 + D2 + 2H}$$  \hspace{1cm} (12)

Where D1 and D2 are the shortest paths from c1 and c2 to c, and H is the shortest path from c to the root.

SEMANTIC SIMILARITY BETWEEN CONCEPTS (LINGUISTIC SIMILARITY MEASURE)
Linguistic Similarity [4] uses Synonym set from WordNet to calculate the similarity between two concepts from different ontologies. Linguistic Similarity is calculated using:

$$S = \min \left[ \frac{c_a}{T_a}, \frac{c_b}{T_b} \right]$$  \hspace{1cm} (1)

Let A and B are two different concepts from two different ontologies respectively. LA and LB represents the list of synonyms for the concepts A and B respectively.

$$CA - \text{number of words in the list } \text{LA that are on } \text{LB}$$
$$TA - \text{total number of words in the list } \text{LA}$$
$$CB - \text{number of words in the list } \text{LB that are in } \text{LA}$$
$$TB - \text{total number of words in the list } \text{LB}.$$  

GENERATION OF FUZZY RULES
Based on two different similarity measure, using Structural similarity, concepts are classified as low similar and highly similar concept as shown in the Fig 2, in the same manner again the concepts are classified as low similar and highly similar concepts using Semantic similar concepts as shown in the Fig 3. By applying fuzzy rules, four different fuzzy rules are generated.

**Rule 1:** IF (path IS low) OR (linguistic IS low) THEN Output IS verylow

**Rule 2:** IF (path IS low) OR (linguistic IS high) THEN Output IS low

**Rule 3:** IF (path IS high) OR (linguistic IS low) THEN Output IS low

**Rule 4:** IF (path IS high) OR (linguistic IS high) THEN Output IS high

MEASURING SEMANTIC RELATEDNESS USING ADABOOST ALGORITHM
The concepts retrieved are further optimized by using AdaBoost Algorithm, which is used to boost the results that are obtained from the Fuzzy System. AdaBoost Algorithm is explained in Fig.5.
To calculate the recall measure, use:

\[
\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

Table 1: Performance of the System (User-defined ontologies)

<table>
<thead>
<tr>
<th>ALGORITHM NAME</th>
<th>PRECISION (%)</th>
<th>RECALL (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy System</td>
<td>88.76</td>
<td>86.36</td>
</tr>
<tr>
<td>AdaBoost Algorithm</td>
<td>92.35</td>
<td>90</td>
</tr>
<tr>
<td>RankBoost Algorithm</td>
<td>93.13</td>
<td>90.16</td>
</tr>
</tbody>
</table>

**Algorithm: AdaBoost**

Given: \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x_i \in X, y_i \in \{-1, +1\}\).

Initialize: \(D_0(i) = 1/m\) for \(i = 1, 2, \ldots, m\).

For \(t = 1, 2, \ldots, T\):

- Train weak learner using distribution \(D_t\).
- Get weak hypothesis \(h_t: X \rightarrow \{-1, +1\}\).
- Aim: select \(h_t\) with low weighted error:
  \(e_t = \sum_{i=1}^{m} \frac{D_t(i)}{m} h_t(x_i) = y_i\).
- Choose \(\alpha_t = \frac{1}{2} \ln \frac{1-e_t}{e_t}\).
- Update, for \(i = 1, \ldots, m\):
  \(D_{t+1}(i) = \frac{D_t(i) \exp(\alpha_t y_i h_t(x_i))}{Z_t}\)

where \(Z_t\) is a normalization factor (chosen so that \(D_{t+1}\) will be a distribution).

**Algorithm: RankBoost**

Given: initial distribution \(D\) over \(X \times X\).

Initialize: \(D_0 = D\).

For \(t = 1, 2, \ldots, T\):

- Train weak learner using distribution \(D_t\).
- Get weak ranking \(h_t: X \rightarrow R\).
- Choose \(\alpha_t \in R\).
- Update:
  \(D_{t+1}(x_0, x_i) = \frac{D_t(x_0, x_i) \exp(\alpha_t (h_t(x_0) - h_t(x_i)))}{Z_t}\)

where \(Z_t\) is a normalization factor (chosen so that \(D_{t+1}\) will be a distribution).

![Fig. 5: AdaBoost Algorithm](image)

Output the final hypothesis:

\[
H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)
\]

**MEASURING SEMANTIC RELATEDNESS USING RANKBOOST ALGORITHM**

Simultaneously, concepts retrieved are further optimized by using RankBoost Algorithm, by which results retrieved from fuzzy system are boosted. RankBoost Algorithm is explained in Fig.6.

![Fig. 6: RankBoost Algorithm](image)

Output the final ranking:

\[
H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)
\]

**IV. PERFORMANCE EVALUATION**

Concepts which are retrieved from the fuzzy system are boosted using AdaBoost and RankBoost algorithm. The retrieved concepts from Fuzzy System, AdaBoost Algorithm and RankBoost Algorithm are compared. The results are obtained and the performance of the system is measured using Precision and Recall.

To calculate the precision,

\[
\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}
\]

To calculate the recall measure,

\[
\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

Table 2: Performance of the System (Benchmark dataset (OAEI))

<table>
<thead>
<tr>
<th>OAEI datasets</th>
<th>ALGORITHM NAME</th>
<th>PRECISION (%)</th>
<th>RECALL (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test#201</td>
<td>AdaBoost Algorithm</td>
<td>93</td>
<td>91.09</td>
</tr>
<tr>
<td>test#222</td>
<td>RankBoost Algorithm</td>
<td>94.5</td>
<td>93.67</td>
</tr>
<tr>
<td>test#303</td>
<td>AdaBoost Algorithm</td>
<td>86</td>
<td>83.89</td>
</tr>
<tr>
<td>test#304</td>
<td>RankBoost Algorithm</td>
<td>92.17</td>
<td>88.76</td>
</tr>
<tr>
<td>test#301</td>
<td>Fuzzy System</td>
<td>85.53</td>
<td>82.07</td>
</tr>
<tr>
<td>test#302</td>
<td>AdaBoost Algorithm</td>
<td>91.09</td>
<td>88.63</td>
</tr>
<tr>
<td>test#303</td>
<td>RankBoost Algorithm</td>
<td>92.39</td>
<td>91.92</td>
</tr>
<tr>
<td>test#240</td>
<td>Fuzzy System</td>
<td>83.13</td>
<td>79.48</td>
</tr>
<tr>
<td>test#247</td>
<td>AdaBoost Algorithm</td>
<td>91.15</td>
<td>83.16</td>
</tr>
<tr>
<td></td>
<td>RankBoost Algorithm</td>
<td>91.91</td>
<td>89</td>
</tr>
<tr>
<td>test#247</td>
<td>Fuzzy System</td>
<td>87</td>
<td>87.16</td>
</tr>
<tr>
<td></td>
<td>AdaBoost Algorithm</td>
<td>91.16</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>RankBoost Algorithm</td>
<td>93</td>
<td>88.16</td>
</tr>
</tbody>
</table>

Average Precision of the Fuzzy System: 86.35
Average Precision of the AdaBoost Algorithm: 91.71
Average Precision of the RankBoost Algorithm: 92.49
Average Recall of the Fuzzy System: 84.51
Average Recall of the AdaBoost Algorithm: 87.32
Average Recall of the RankBoost Algorithm: 89.56

Table 1 depicts the performance of the proposed approach. The performance of the proposed fuzzy system is calculated using Precision and Recall. Figure 4 represents the performance comparison chart.

**V. CONCLUSION AND FUTURE WORK**

User browsing experience can be enhanced by increasing the efficiency of retrieval of relevant document using Query Expansion techniques. The Proposed approach retrieves relevant document by removing vague and imprecise concepts by applying fuzzy rules and to the results boosting algorithms like AdaBoost and RankBoost Algorithm can be used and hence this could be used for optimising searches in domain specific information retrieval system for efficient information retrieval. The result is based on both hierarchical and non-hierarchical relationships of concepts in the ontology.

**REFERENCES**


