

Semi Automatic Ontology Based Bilingual Information Retrieval System (Pilgrimage Tourism in South India)

Dr. S. Saraswathi¹

Pondicherry Engineering College, Pondicherry, India

Email: swathi@pec.edu

Jemibha P, Sugandhi M, Mathimozhi M, Lourdu Sophia A²

Pondicherry Engineering College, Pondicherry, India

Email: jemibha91@gmail.com, sugan2910@gmail.com, mathilucky91@gmail.com, sophiasebastin@gmail.com

A. Nagarathinam³

Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, India

Email:rathna_chennai01@yahoo.co.in

Abstract— This paper focuses on the construction of a Semi Automatic Ontological tree in the domain of Pilgrimage Tourism in South India for the purpose of enhancing the efficiency in the online Information Retrieval. The proposed system uses two languages Tamil and English for the input query and document retrieval. The user can pose the query in either Tamil or English and the resultant document will be displayed in the query language. In order to retrieve more relevant documents, a semi-automatic Ontology tree has been constructed. The semi automatic ontological tree uses only the English language. Machine Translation approach is used to translate the retrieved result to the language that of the user's query. Our system produces the better results for the simple user's query about Pilgrimage Tourism in South India for which the answers could be retrieved from the updated semi automatic ontological tree itself.

Index Terms—Semi automated ontology, Ontology reusability, Ontology overload

I. INTRODUCTION

The main research area in Semantic Web is the Ontology construction. In general, the term Ontology refers entities and their relationships for a given domain. Ontology is used in query expansion process thereby more related terms could be added with the given search keywords which are later used to retrieve more relevant documents in information retrieval. For a pre-specified domain, it is very tedious to construct or update the ontology manually as it consumes more man power and time. Since the ontology is going to be involved in the Web Search related task, it is always better to reconstruct the Ontology from the relevant Web pages. Thereby, we can reduce the Overload of managing the Ontology.

Semantic Web will not be a new global information highway parallel to the existing World Wide Web; instead it will gradually evolve out of the existing Web [1]. Ontologies are built in order to represent generic knowledge about a target world [2]. Gruber defined

ontology as a specification of a conceptualization [3]. Ontology defines the basic terms and their relationships comprising the vocabulary of an application domain and the axioms for constraining the relationships among terms. This definition explains what an ontology looks like [4-5].

Reconstructing the Ontology from the relevant Web pages also reduces the testing process and enhances the mapping process between different documents in Information Retrieval.

In this paper, we discussed our work under many sections. In the first section, the literature review of Semi-automated ontology construction has been specified to identify the essentials of Ontology reusability. The second and third section deals with our way of constructing the ontology. Finally, we analyzed our work and have shown the enhancement in the performance of Ontology in Web search. The proposed system handles two languages. English, the most globally used language. Most of the documents in online search are in English. The second language is Tamil. Tamil is a South Indian language spoken widely in Tamil Nadu in India. Tamil has the longest unbroken literary tradition amongst the Dravidian languages.

II. RELATED WORK

Ontology is one among the concept for which researches are been undertaken worldwide in order to extract efficient use from it in the design of web search Engine. The related work for our proposed concepts is discussed in detail by the following researchers in their research papers.

Uschold et al.,[6] experimented the reusability concept in constructing the ontology in a small-scale application and concluded that reusing an ontology was remote from an automated process.

Noy and Musen [7] specified that the users can extract a part of ontology from the given domain ontology by following the traversal view of the ontology.

Stuckenschmidt and Klein [8] used the structure-based algorithm to extract a specific part of ontology from a given large Ontology.

Pinto and Martins [9] created large domain ontology by means of assembling, extending and integrated tiny predefined ontologies.

Deryle Lonsdale et al., [10] used concept selection, relation retrieval and constraint discovery for reusing an automated ontology. They used a large-scale ontology named MikroKosmos (μ K) which was in XML DTD format. Also, they used their own lexicon base for their domain, WordNet and a set of declarative data recognizers for their reusing process.

Some set of seed concepts have been used to identify their related concepts or new patterns using synonyms from the Web [11] by considering a new w patterns for the given seed words.

OntoLearn [12] has incorporated both symbolic and statistical methods in its Ontology learning system. It used text documents in English as the input and WordNet as the resource for ontology enrichment. It checked the relationship between the concepts in Ontology using a set of Domain specific terms and a rule-based inductive learning method.

DODDLE II [13] used a machine-readable dictionary and domain-specific texts. Matching and trimming methods with WordNet have been used to identify the taxonomic relationships; whereas, domain-specific text along with WordSpace[14] have been used to analyze the non-taxonomic relationships. In this, a statistical co-occurrence information detail has been identified.

Yuchul Jung et al., [15] proposed automatic Ontology construction by using some set of "How-to" instructions from the large scale Web resources namely eHow[16], a free online community which allows the researcher to view and share their views, and WikiHow[17], a community based web site having enormous database with how-to instructions. Based on the situation awareness, the situation domain has been constructed. Preprocessing, syntactic pattern-based approach along with probabilistic CRF –based approach were used to construct the Ontology from the Web resources.

III. MODULES OF THE PROPOSED SYSTEM

Usage of the Semi Automatic Ontological tree is the striking feature of our system. A Semi Automatic Ontological tree can be defined as entities and grouping of entities, classification of entities in a hierarchical order and grouping of entities based on similarities and differences where the entities are created partially manual and it is completed dynamically. This kind of dynamic updation in the manually constructed Ontological Tree normally helps in enhancing the efficiency of the concept words retrieved from the Ontological Tree for its usage in the Retrieval of Information relating to the User's Query for search.

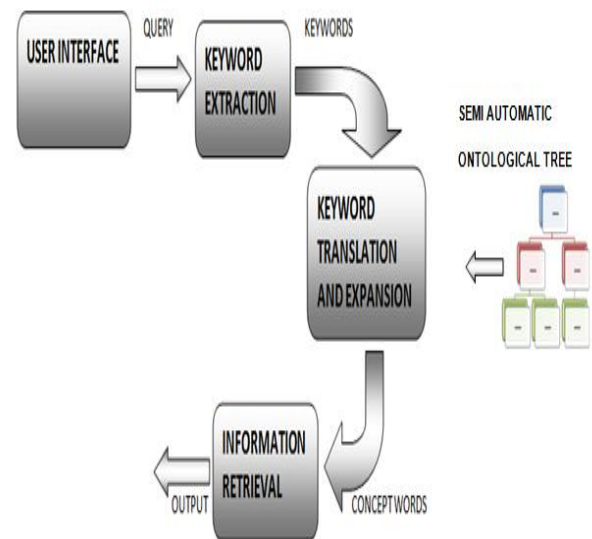


Figure 1. Information Retrieval Using Semi-Automatic Ontology

The Semi Automatic Ontological tree is used in the phase of dynamically completing the Ontology tree which was initially created manually for effective retrieval of the related keywords or concept words for the given query.

We have created a Semi-Automatic Ontology tree consisting of four levels which includes State, District, Pilgrimage place and the last level consists of the attributes related to each Pilgrimage place. In this system the attributes chosen are the Name of the Deity worshiped, Place in which it is situated, Bus Facility if available, Train Facility if available and Airport Facility if available.

Due to lack of efficiency in constructing the tree manually, the created tree may have some missing information. The missing information could be about the selected attribute value of a particular Pilgrimage or the Pilgrimage itself or a District under a State.

Such missing information is dynamically filled through online search as a first stage of the system. This makes the tree to be effective in its use.

The semi automatic ontological tree constructed can also be used to retrieve result for certain simple queries by the user.

3.1. Building of Semi Automatic Ontological Tree:

For this process, we use XML for building of a semi automatic ontological tree. It satisfies a number of constraints as listed below:

- Nodes with single entry.
- Dynamic update of Node values.
- Single Level with multiple attribute value.

XML (Extensible Markup Language) is a user defined Language which is predominantly used for various domains across the globe. Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable[18]. The design goals of XML emphasize simplicity, generality,

and usability over the Internet[19]. It is a textual data format with strong support via Unicode for the languages of the world.

In XML language general tree construction is effective and easy because of its flexibility. Retrieval of attribute value from a XML tree is also more time efficient which is essential for the proposed system. Thus the semi automatic ontological tree for this system is constructed in XML. The sample overview of the manually constructed semi automatic ontological tree used for this domain is as follows:

```

<PILGRIMAGES>
  <TamilNadu>
    <Chennai>
      <temples>
        <NAME>sivan</NAME>
        <GOD>siva</GOD>
        <LOC>ariyalur</LOC>
        <BUS>ariyalur</BUS>
        <TRAIN>ariyalur</TRAIN>
        <AIR>ariyalur</AIR>
      </temples>
      <temples>
        <NAME>St. Mary's Church </NAME>
        <GOD>Mary</GOD>
        <LOC>Anna Salai Road</LOC>
        <BUS>Egmore</BUS>
        <TRAIN>Central</TRAIN>
        <AIR>Meenambakkam</AIR>
      </temples>
      <mosques>
        <NAME>Thousand Lights
          Mosque</NAME>
        <GOD>Allah</GOD>
        <LOC>Anna Salai Road</LOC>
        <BUS>Egmore</BUS>
        <TRAIN>Central</TRAIN>
        <AIR>Meenambakkam</AIR>
      </mosques>
    </Chennai>
    <Coimbatore>
      <temples>
        <NAME>murugan</NAME>
        <GOD>muruga</GOD>
        <LOC>chennai</LOC>
        <BUS>chennai</BUS>
        <TRAIN>chennai</TRAIN>
        <AIR>?</AIR>
      </temples>
      <temples>
        <NAME>Perur Patteeswara Swamy Temple
</NAME>
        <GOD>Lord Shiva</GOD>
        <LOC>river Noyyal</LOC>
        <BUS>Coimbatore</BUS>
        <TRAIN>Coimbatore Junction</TRAIN>
        <AIR>Coimbatore</AIR>
      </temples>
      <temples>
        <NAME>Anubhavi Subramaniam
Temple </NAME>
        <GOD>Lord Subramania</GOD>
        <LOC>Sathyamoorthy Road, Ramnagar of
Coimbatore city</LOC>
        <BUS> Coimbatore </BUS>
        <TRAIN> Coimbatore Junction</TRAIN>
        <AIR>Coimbatore</AIR>
      </temples>
      <temples>
        <NAME>Ayyappan temple </NAME>
        <GOD>Ayappan</GOD>
        <LOC>Coimbatore</LOC>
        <BUS>Coimbatore</BUS>
        <TRAIN>Coimbatore Junction</TRAIN>
        <AIR> </AIR>
      </temples>
      <temples>
        <NAME>Ramar temple</NAME>
        <GOD> </GOD>
        <LOC>Coimbatore</LOC>
        <BUS>Coimbatore</BUS>
        <TRAIN>Coimbatore Junction</TRAIN>
        <AIR>Coimbatore</AIR>
      </temples>
    </Coimbatore>
    <Nagapattinam>
      <church>
        <NAME>Velankanni </NAME>
        <GOD>Mother Mary</GOD>
        <LOC>Velankanni </LOC>
        <BUS>Velankanni </BUS>
        <TRAIN>Nagapattinam </TRAIN>
        <AIR>Thiruchirapalli</AIR>
      </church>
      <temples>
        <NAME>Sikkal Singaravelar
Temple</NAME>
        <GOD>Murugan</GOD>
        <LOC>Sikkal</LOC>
        <BUS>Nagapattinam</BUS>
        <TRAIN>Nagapattinam</TRAIN>
        <AIR>Thiruchirapalli</AIR>
      </temples>
    </Nagapattinam>
  </TamilNadu>
  <Kerala>
    <Idukki>
      <Alappuzha>
        <temples>
          <NAME>Mullakkal Rajeshwari Temple
</NAME>
          <GOD>Rajarajeshwari</GOD>
          <LOC>Alappuzha</LOC>
          <BUS>Mullakkal</BUS>
          <TRAIN>Alappuzha</TRAIN>
          <AIR>Thiruvananthapuram</AIR>
        </temples>
      </Alappuzha>
    </Kerala>

```

```

<AndhraPradesh>
  <Adilabad>
    <temples>
      <NAME>Basar Saraswati
      Temple</NAME>
      <GOD>Gnana Saraswati</GOD>
      <LOC>Basar</LOC>
      <BUS>Basar</BUS>
      <TRAIN>Basar</TRAIN>
      <AIR>Tirupati</AIR>
    </temples>
  </Adilabad>
</AndhraPradesh>
</PILGRIMAGES>
    
```

3.2 Design and Implementation of Semi Automatic Ontological Tree:

A) Manual creation of the Ontological tree:

- The details about various districts in the South Indian States such as Tamil Nadu, Karnataka, Andhra Pradesh, Kerala and Pondicherry and Pilgrimage places present in the districts and the details about each Pilgrimage place are collected through online search.
- Such details are consolidated and only required information such as the name of the pilgrimage place, its location, the deity worshipped, and other facilities to reached the place is used for the construction of the ontological tree.
- A XML tree is constructed using such details which may also consist of some missing or incomplete or irrelevant information to the corresponding node.

B) Dynamic update:

- During this process of dynamic updation of the manually constructed Ontological Tree each level of the Ontological Tree is checked for its completeness using the details searched in online dynamically.
- In the case of finding incomplete or missing information or irrelevant information, such information is filled dynamically through retrieving related documents from online search.

The following algorithm is used to update the Semi-automatic ontology in a dynamic way.

C) Algorithm for Semi-Automatic Ontology Construction:

1. Retrieve an online document to check the second level of the tree that is to check the presence of all the districts under each state in the manually constructed Ontological Tree.
2. If any district is found missing under any state in the constructed ontological tree, then the Pilgrimage details about that particular district is added under the corresponding tree.
3. The attribute details of the Pilgrimage place in the missing districts are searched in online and the relevant details retrieved are consolidated.
4. Such details are then added to the Ontological tree dynamically.
5. Search for the completeness of the attribute details of the temples.

6. If any attribute value of a particular pilgrimage place is incomplete, then such information is also retrieved from online documents and it is added.

7. This forms the semi automatic ontology tree to be used for various purposes such as concept words expansion and answering of simple queries in the proposed system

IV. MODULES OF SEMI-AUTOMATIC ONTOLOGY BLIR SYSTEM

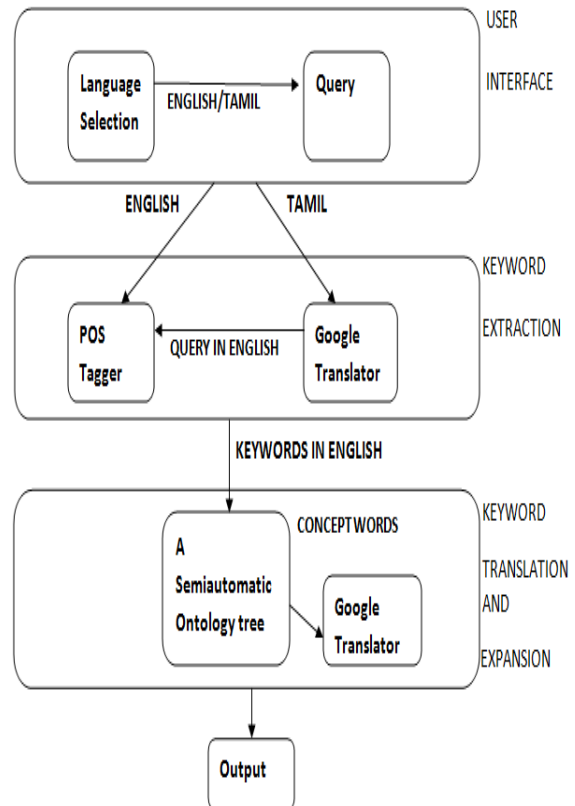


Figure. 2. Overall Module Diagram of the proposed system

Module 1: User Interface Design

The first stage of our proposed system implementation is User Interface Designing. The main role of this module is to get a query from the user whose solution is to be found. The query can be of any language among the two languages (Tamil and English). The user interface is designed in Java using AWT objects and classes.

The various sub modules of UI are as follows

A) Language selection:

The user is initially asked to choose one among the two languages namely: Tamil or English to enter query. The user can select a language in which he / she are most comfortable. The output of the query is also expected to be in the language chosen by the user.

B) Input query:

When the user is finished with selecting the language in which he / she wants to give a query, the query can be

entered in the place provided for it. A virtual Keyboard which is appropriate to the selected language is used to facilitate the user to give his query as input.

Module 2: Keyword Extraction

Input: User's query

Output: Keywords and the related tagger information from the input query.

A) Tagger Software

A Tagger is a software which is used to take a sentence as input, separate the words in that sentence, identifies the parts of the sentences separated like nouns, verbs, preposition etc. and returns every word in the sentence along with its type as an output.

The Query given by the user is fed as an input to the appropriate Tagger. It tags the input sentence and returns the parts of the sentences. From the output of the tagger, verbs and nouns are identified and are set as main keywords to perform the document search. The type of Tagger used in this system is Win Tree Tagger. In case of query in Tamil, the query is first translated to English using a translator. Then the translated query is given as input to the Win Tree Tagger and it is tagged.

In case of Tamil Query, the query is first translated to English using a Translator (Google Translator in our implementation) and then the query is tagged using the Win Tree Tagger.

B) Identification of simple vs. complex query:

Simple query is a query by the user that can be answered with the details present in the ontological tree. Such queries are identified after the tagging process and are processed separately using the semi automatic ontological tree. In our implemented system only simple query is assumed to be given as input to the system by the user. This makes the system more efficient by the usage of Semi Automatic Ontological Tree.

Module 3: Keyword Extraction and Translation:

The objective of this module is to extract related keywords and concepts words for the query by the user using the semi automatic ontological tree. The concept words from the Semi Automatic Ontological Tree are provided to the user as the result of the Query which is given by him.

Usage of Semi Automatic Ontological Tree:

Each keyword identified is matched with entry in every node in the Semi Automatic Ontological tree. The exact location of the keyword in the tree is identified. All the child nodes of the keyword node are traversed and their corresponding entries are noted as related data for document search.

Module 4: Information Retrieval

The related data retrieved from Semi Automatic Ontological tree is given as the output to the user in the

query language. If the query is in Tamil then translation has to take place before presenting the result.

A) Machine Translation

Machine Translation (MT) is the process of using computer software to translate text or speech from one natural language to another [20]. The language in which the solution is to be retrieved is checked with the user chosen language. If it is found to be the same, the solution is directly presented to the user. Else, Machine Translation is done to re convert the solution that is found back to the user's query language and then presented to the user.

V. PERFORMANCE ANALYSIS OF SEMI-AUTOMATIC ONTOLOGY BLIR SYSTEM

Parameters for Testing:

1. Precision: Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search [20].

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad (1)$$

2. Recall: Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents [20].

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad (2)$$

Various types of queries are given as input to the implemented system from the perspective of a user and the corresponding results are analyzed using both a manually constructed Ontology and a Semi-Automatic Ontology in the pilgrimage domain.

We have examined five types of queries namely,

Type A - Details about the pilgrimage place in a particular district.

Type B - Details about the deity worshipped in any pilgrimage place.

Type C - Questions related to the location of a pilgrimage.

Type D - Pilgrimage place where a particular deity is worshipped.

Type E - Bus/Train/Air facility available to reach a pilgrimage place.

More than 20 questions for each type have been given as input and the output is obtained from the implemented proposed system and the corresponding result is analyzed. Some of the example queries are:

- Which airport is nearer to reach Tirupati Venkatachalapathy Temple?
- Which Railway station is nearer to St John Church in Secundarabad?
- How to reach Mohammadiya Mosque in Chikballapur by road?
- List the famous Durga temples in South India

- List the mosques in Kerala
- Which God is worshipped in Padmanabha Temple in Kerala?
- List the CSI churches found in Pondicherry
- What is the way to reach Omkareshwara temple?
- List the famous places of worship at Chennai.
- List the temples at Adilabad where Lord Shiva is worshipped as deity.
- List the details about Meenakshi Temple located at Madurai
- What are Churches where Mary is worshipped at Andhra Pradesh District?
- What Roman Catholic Churches present in Nagapattinam district?

For all the five types of queries, our System will examine the Semi-Automated Ontology. As we have already stated, in our Ontology, we have also included the missing information during the construction of Ontology which can be seen in the sample shown in this paper.

Once the user has entered the query, our system will find the keywords and try to extract the relevant concept words for the given keywords from the existing Ontology. If the relevant information is present in the Ontology itself then those concept words would be used to extract the information from the Ontology. On the other hand, if the Ontology contains some missing information for the given pilgrimage then from the Search Engine (we used Google) the corresponding details will be retrieved and updated in the Ontology. Later, we can get the relevant answers for the user's query from the Semi-Automated Ontology.

We used the performance measures namely Precision and Recall to analyze the working efficiency of our proposed system. As mentioned the term Precision refers the number of relevant documents retrieved from the total number of documents and Recall refers the number of relevant documents retrieved from the total relevant documents.

Table 1 shows comparison of Precision for our proposed system. From the analysis, we can identify that the number of relevant documents retrieved for the given query has been increased when we are using Semi-Automated Ontology. We can notice that for the Type A, B, C and D, the Precision is high in our system compared with that of manually created Ontology. For the Type E the results of our system are same as that of manually constructed Ontology.

The statistics clearly shows that our System gives better results for the user's query related to details about the pilgrimage places in a particular district, details about the deity worshipped in any pilgrimage place, general details of the location and the details about deities.

Table 2 reveals that Recall is also high in our system, compared with the system using manually constructed ontology. Only for Type C queries, there is an equal Recall factor for the manually constructed and semi automatically constructed Ontology. This is because the location of a particular pilgrimage place remains the same always. Therefore, there is no possibility of change in its

value while updating the Ontological Tree. For the remaining query types such as A, B, D and E, we can see the increased Recall measures with our System.

Figure 3 and 4 shows the pictorial view of the analysis of the system with respect to the Precision and Recall measures. In summary, it is found that both the Precision and Recall are better for a system using semi automated Ontology compared with that using manually constructed Ontology.

The results exposes that the number of relevant documents retrieved for the queries of Type A, B and D is high when we are using a Semi-Automated Ontology. Since the details about the bus or train or air facility for a particular pilgrimage place is hard to vary, it takes the same value after the dynamic updation process. Therefore, there is no much difference in the performance of our system while we are retrieving the documents for queries of Type C and E. However the overall performance of the system is better when we are using a Semi-Automatic Ontology in Bilingual Information Retrieval system.

Table 1. Comparison of Precision

Type of Query	Manually created Ontological Tree	Semi Automated Ontological Tree
A	0.698	0.724
B	0.712	0.719
C	0.675	0.682
D	0.677	0.691
E	0.684	0.684

Table 2 Comparison of Recall

Type of Query	Manually created Ontological Tree	Semi Automated Ontological Tree
A	0.815	0.826
B	0.799	0.812
C	0.834	0.834
D	0.801	0.817
E	0.821	0.833

VI. CONCLUSION

The proposed system is found to be efficient for all the simple queries which can be answered from the updated semi automatic ontological tree. The user can ask the query in either Tamil or English and the resultant documents will be displayed in the query language. Thereby the user can get the result in non-English also.

The process is time efficient since the manual construction of ontological tree is a onetime process and the process of updating the manually constructed ontological tree followed by result retrieval takes less time compared to the consolidation of the result from the online search.

We have used Semi-Automatic ontology in Pilgrimage domain of South India. The results show that there is a significance improvement in the Precision and Recall of Bilingual Information Retrieval when we are using Semi-Automated Ontology. The automatic updating in Ontology not only improves the performance of document retrieval but, also facilitates the researcher to have some missing information in their domain which can later be filled by searching the content from the Web. Thus the ontology construction for a particular domain becomes easier for the researcher. The future enhancement can be extending this system to use complex query or query where details need result in a summarized format.

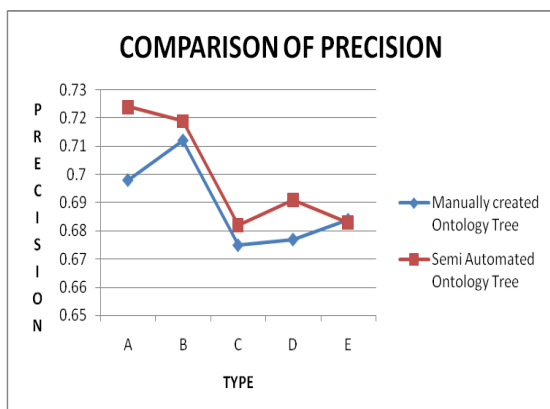


Figure 3. Analysis of Semi-Automatic Ontology based BLIR System in terms of Precision

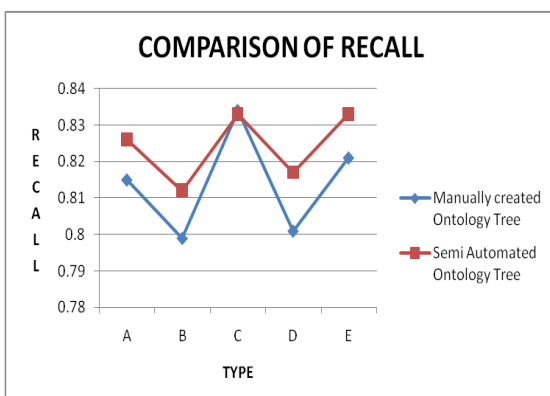


Figure 4. Analysis of Semi-Automatic Ontology based BLIR System in terms of Recall

REFERENCES

1. Grigoris Antoniou and Frank van Harmelen, *A Semantic Web Primer*, PHI Learning Private Limited, New Delhi, 2010, pp 1-3
2. M Bung, *Treatise on Basic Philosophy. Ontology I. The Furniture of the World. Vol. 3*, Boston: Reidel.
3. T Gruber, *What is an Ontology?* (September, 2005): <http://www.ksi-stanford.edu/kst/what-is-anontology.html>.
4. X Yang, *Ontologies and How to Build Them* 2001. (March, 2006) <http://www.ics.uci.edu/~xwy/publications/area-exam.ps>.
5. D. Bugaite, O. Vasilecas. *Ontology-Based Elicitation of Business Rules*. In A. G. Nilsson, R. Gustas, W. Wojtkowski, W. G. Wojtkowski, S. Wrycza, J. Zupancic *Information Systems Development: Proc. of the ISD'2004*. Springer-Verlag, Sweden, 2006, pp. 795-806.
6. M. Uschold, M. Healy, K. Williamson, P. Clark, S. Woods, *Ontology reuse and application*, in: *Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS'98)*, Trento, Italy, 1998, pp. 179-192.
7. N. Noy, M. Musen, *Specifying ontology views by traversal*, in: *Proceedings of the Third International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 2004, pp. 713-725.
8. H. Stuckenschmidt, M. Klein, *Structure-based partitioning of large class hierarchies*, in: *Proceedings of the Third International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, 2004, pp. 289-303.
9. H. Pinto, J. Martins, *Reusing ontologies*, in: *AAAI Spring Symposium on Bringing Knowledge to Business Processes*, 2000, pp. 77-84, Stanford University, California, March 20-22.
10. Deryle Lonsdale, David W. Embley, Yihong Ding, Li Xu, Martin Hepp, *Reusing ontologies and language components for ontology generation*, *Data & Knowledge Engineering* 69 (2010) pp. 318-330
11. Kaihong Liu, William R. Hogan, Rebecca S. Crowley, *Natural Language Processing methods and systems for biomedical ontology learning*, *Journal of Biomedical Informatics* 44 (2011) pp. 163-179
12. Velardi P, Navigli R, Cucchiarelli A, Neri F. *Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies*. In: *Proceedings of ECAI and EKAW, 2004*
13. Yamaguchi T. *Acquiring conceptual relationships from domain-specific texts*. In: *Proceedings of IJCAI Workshop on Ontology Learning (OL)*. USA: Seattle; 2001.
14. Hearst MA, Schutze H, *Customizing a lexicon to better suit a computational task*. In: *Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, Columbus, OH, 1993
15. Yuchul Jung, Jihee Ryu, Kyung-min Kim, Sung-Hyon Myaeng, *Automatic construction of a large-scale situation ontology by mining how-to instructions from the web*, *Web Semantics: Science, Services and Agents on the World Wide Web*, Volume 8 (2010), pp. 110-124
16. <http://www.ehow.com>
17. <http://www.wikihow.com>
18. <http://en.wikipedia.org/wiki/XML>

19. "XML Media Types, RFC 3023". IETF. 2001-01. pp. 9–11. Retrieved 2010-01-04.

20. Dr. S. Saraswathi, Asma Siddhiqaa. M, Kalaimagal. K, Kalaiyarasi.M, "BiLingual Information Retrieval System for English and Tamil", Journal of Computing, Volume 2, Issue 4, April 2010, ISSN 2151-9617.

How to cite this paper: S. Saraswathi, Jemibha P, Sugandhi M, Mathimozhi M, Lourdu Sophia A.A. Nagarathinam, "Semi Automatic Ontology Based Bilingual Information Retrieval System (Pilgrimage Tourism in South India)", International Journal of Intelligent Systems and Applications (IJISA), vol.4, no.3, pp.48-55, 2012. DOI: 10.5815/ijisa.2012.03.07



First A. Author S. Saraswathi is Assistant professor, in the Department of Information Technology, Pondicherry Engineering College, Pondicherry, India. She completed her PhD, in the area of speech recognition for Tamil language. Her areas of interest include speech processing, artificial intelligence and expert systems.



Second B. Author Jr Jemibha P is final year student of Department of Information Technology, Pondicherry Engineering College, Pondicherry, India. Her areas of interest are Artificial Intelligence, Database Management and Data Structures.



Second B. Author Jr Mathimozhi M is final year student of Department of Information Technology, Pondicherry Engineering College, Pondicherry, India. Her areas of interest are Artificial Intelligence, Object oriented Programming concepts and Database Management.



Second B. Author Jr Sugandhi M is final year student of Department of Information Technology, Pondicherry Engineering College, Pondicherry, India. Her areas of interest are Artificial Intelligence, Data Structures and Operating System.



Second B. Author Jr Lourdu Sophia A is final year student of Department of Information Technology, Pondicherry Engineering College, Pondicherry, India. Her areas of interest are Artificial Intelligence, Object Oriented Programming Language and Computer Networks.



Third C. Author Jr Nagarathinam A Tamilnadu, India, Associate Professor and pursuing Ph. D in Manonmaniyam Sundaranar University. Her research interest includes Information Retrieval and Natural Language Processing.