CHAPTER 4

TOPIC ONTOLOGY CONSTRUCTION BASED ON WIKIPEDIA AND WORDNET

4.1 NEED FOR SEMANTIC WEB

Due to the unexpected growth of World Wide Web (WWW), vast amount of information are available over the internet. In order to access the web resources, a large number of standard web mining algorithms and information retrieval techniques have been designed. However they rely only on simple keyword matching. Therefore, an user can meet the situation of inconvenience while accessing the web resources because the user does not consider semantic concepts between the web content in large corpus of documents (Fortuna et al 2005). However from 1990’s, most of the researches paid their attention on Semantic Web which relies on the evolution of ontology concepts. These concepts describe the conceptual relationship between entities in a specific domain with efficient knowledge sharing and machine interoperability. Ontologies are simply defined as the taxonomy of hierarchy of concepts. It is a way of organizing and representing the entities and associations of a conceptual model in a domain in conjunction with semantic relationships. It allows people to communicate with computers via shared resources within a domain. It is mainly constructed to provide the knowledgeable representation that can describe the web resources using intelligent techniques understandable by human and processable by machine (Sanchez and Moreno 2004). In ontology, concepts in specific domain are formulated using proper encoding mechanism that can support efficient
information retrieval. High information load due to the large corpus of documents is reduced.

![Figure 4.1 Semantic Web Architecture as Presented by Tim Berners-Lee](image)

The main goal of the Semantic Web is to enhance the interoperability among web systems by describing information on the Web in a semantically meaningful method. Therefore, the World Wide Web Consortium (W3C) Semantic Web activity defines a set of standards, which are arranged in a layered architecture as shown in Figure 4.1. Resources on the Semantic Web are identified via a Uniform Resource Identifier (URI) and are depicted using the Resource Description Framework (RDF). RDF Schema and the Web Ontology Language (OWL) allow for the specification of Ontologies.

### 4.2 TOPIC ONTOLOGY

Topic ontology is defined as the ontology with set of topics that are interconnected using semantic relations (Stamou and Ntoulas 2009). It is also defined as the graph in which each node represents the specific topic that
forms a topic hierarchy. It comprises a group of relevant topics related to the specific concept. A hierarchical semantic relationship is maintained among the concepts in the topics. Topic ontology can be constructed in many ways but it needs to be ordered in a hierarchy fashion. The construction process extracts topical keywords from the text corpus using any standard text mining and information retrieval techniques and builds topic ontology based on semantic relevance of the keywords. In the recent years, many researches focused on the creation of ontology manually which comprises of scope, reuse, hierarchy and instances. The keyword based construction approach is not efficient because it is not possible to construct ontology from the large corpus of web documents (Maguitman et al 2010).

4.3 EXISTING ONTOLOGY CONSTRUCTION APPROACHES

With more and efficient usage of ontology in a wide variety of applications it becomes a dire necessity to frame a suitable construction approach for ontology. Ontologies are constructed using two methods such as i) manual process (semi automated) with supervised approach ii) Automated process with unsupervised approach. As of today, construction approaches for most of the applications deal with a semiautomated process. Ontologies are constructed manually using various approaches such as data mining techniques, machine learning techniques, NLP methods that can construct ontology with high quality. But in the aspect of performance it leads to a very expensive, time consuming and also vary hard to maintain the constructed ontology. Besides, the difficulty in maintenance of the same makes this approach less feasible for a wide degree of applications. On the other hand, automated approach can construct ontology without the effort of user intervention. However, several problems exist in this unsupervised approach like noisy, inconsistency, proper handling of missing information of concepts and their relationships deserve a proper mention. Added to this, the factor of
inconsistency plays a key role in diminishing its effectiveness. Therefore, the construction of high quality and efficient ontology is still remains as an open research problem. In order to overcome the limitation in already available construction process, an automated topic ontology approach has been proposed.

4.4 TOWARDS AN AUTOMATIC TOPIC ONTOLOGY CONSTRUCTION APPROACH

In order to effectively implement the proposed approach, an ideal ontology needs to be identified. ODP is found to be a suitable base due to the fact that it contains all necessary information.

The proposed work is the first approach that takes one step forward to construct topic ontology through enriching the set of categories in existing small ontology called ODP. In this work, a novel technique based corpus approach has been proposed to enrich the set of categories in ODP through automatically identify the concepts and their associated semantic relationship from corpus based external knowledge resources such as wikipedia and WordNet. The core concept of this research is that semantic relation derived from the WordNet is used to enrich the automatic construction of topic ontology using a set of highly interlinked topic categories extracted from Wikipedia pages (Grobelnik and Mladenic 2005).

This approach is implemented on open source platform using Netbeans 6.8 developing environment of Java programming language J2EE (JSP servlets). The proposed work uses the Tomcat 5.5 server to deploy the various necessities that deal with the web applications. Concretely, the tools used in this work are described as follows: Html Parser 1.4: This is a powerful HTML parser that allows parsing the required information from the online
encyclopedia, wikipedia. The complete structure of this approach has been depicted in Figure 4.2.
4.4.1 Open Directory Project (ODP)

Open Directory Project (ODP) is a multilingual open content directory of World Wide Web links. The main purpose of ODP is to list out the set of categories related to specific concept. ODP is the largest, most comprehensive human-edited web directory that offers a flat categorization of a specific set of categories (Zhu and Dreher 2009). Each of the categories in ODP contains the related Topics, list of sub Categories, list of web pages related to the category and also contains the description for the category. It consists of nearly 4.59 million submitted Web sites, 82,929 editors and 590,000 categories. Entries of ODP are increasing continuously due to the explosive growth of web applications. A hyperlink based approach intended to describe the process on how to enrich the set of categories in existing ontology (ODP) using the semantic concepts (topics) and their corresponding semantic relation extracted from the external knowledge web resources such as Wikipedia and WordNet has been proposed.

![ODP Diagram]

**Figure 4.3 ODP with Set of Categories related to the Bank Concept**
Figure 4.3 shows the ODP for the bank concept in Business category. It consists of a set of topics related to the bank concept such as Card, Account, Loan and other relevant transactions. Table 4.1 shows an example of categorized results for topic ontology taxonomy from the Open Directory Project (ODP). It performs efficient categorization and each category is occupied by more relevant set of topics related to the specific concept.

**Table 4.1 Categorized results for Topic Ontology**

<table>
<thead>
<tr>
<th>Topics</th>
<th>Categorized results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecommerce</td>
<td>1. Online Banking</td>
</tr>
<tr>
<td></td>
<td>▶ Fund transfer</td>
</tr>
<tr>
<td></td>
<td>▶ Bill Payment</td>
</tr>
<tr>
<td></td>
<td>▶ Investment loan</td>
</tr>
<tr>
<td></td>
<td>2. Online shopping</td>
</tr>
<tr>
<td></td>
<td>▶ Logistics</td>
</tr>
<tr>
<td></td>
<td>▶ Product delivery</td>
</tr>
<tr>
<td></td>
<td>▶ Shopping cart</td>
</tr>
<tr>
<td>Financial Services</td>
<td>1. Banking</td>
</tr>
<tr>
<td></td>
<td>▶ account</td>
</tr>
<tr>
<td></td>
<td>▶ Loan</td>
</tr>
<tr>
<td></td>
<td>▶ card</td>
</tr>
<tr>
<td></td>
<td>2. Insurance</td>
</tr>
<tr>
<td></td>
<td>▶ Health Insurance</td>
</tr>
<tr>
<td></td>
<td>▶ Property Insurance</td>
</tr>
<tr>
<td></td>
<td>▶ Life Insurance</td>
</tr>
<tr>
<td>Commercial Enterprise</td>
<td>1. Management</td>
</tr>
<tr>
<td></td>
<td>▶ Human Resource</td>
</tr>
<tr>
<td></td>
<td>▶ Marketing</td>
</tr>
<tr>
<td></td>
<td>▶ Information</td>
</tr>
<tr>
<td></td>
<td>2. Manufacturing</td>
</tr>
<tr>
<td></td>
<td>▶ Textile Industry</td>
</tr>
<tr>
<td></td>
<td>▶ Chemical Industry</td>
</tr>
<tr>
<td></td>
<td>▶ Construction</td>
</tr>
</tbody>
</table>
4.4.2 Wikipedia

Wikipedia is a multilingual online encyclopedia which contains articles and categories that are related to one precise concept. Wikipedia is widely used by the Internet users. In wikipedia, each web page defines the set of topics related to the specified concept in a more formally organized hierarchical fashion. In addition, Disambiguation between similar words is removed by providing the web pages according to the different word senses. The searching in Wikipedia is a straightforward approach. Therefore a set of keywords associated with each web page can construct topic ontology but keywords themselves not enough to construct topic ontology because keywords are just arrangement of set of relevant topics related to specific concept and they do not render any semantic relation among the set of concepts (Nakayama et al 2008).

A open source toolkit for mining Wikipedia, an open-source software system that allows researchers and developers to integrate Wikipedia’s rich semantics into their own applications. The toolkit creates databases that contain summarized versions of Wikipedia’s content and structure, and includes a Java API to provide access to them. Wikipedia’s articles, categories and redirects are represented as classes, and can be efficiently searched, browsed, and iterated over. Advanced features include parallelized processing of Wikipedia dumps, machine-learned semantic relatedness measures and annotation features, and XML-based web services. Wikipedia Miner is intended to be a platform for sharing data mining techniques (Milne and Witten 2013).

4.4.3 WordNet

WordNet is a lexical reference system that consists of set of related usual language terms that falls under same lexical concepts. Each word
possesses one or more similar synonyms depending upon its word sense such as nouns, verbs, adjectives and adverbs. A collection of words that distributes one sense is known as synset and it is organized in the form of hierarchy manner (Miller 1995). The set of related terms in synset possess Hyponym/Hypernym (subclass/superclass, represented as IsA relation) and Meronym/Holonym (part-whole relationships) which is used to facilitate the efficient construction of topic ontology. Figure 4.4 shows the partial hierarchical structure of the tag “Java” based on the relations of WordNet.

Figure 4.4 Partial Hierarchical Structure of the tag “Java” based on the relations of WordNet
The open source toolkit’s extraction package is responsible for gathering summary data from Wikipedia’s XML dumps. The extraction process exploits Hadoop, an open source implementation of Google’s proprietary GFS file system and MapReduce technology. The former implements a reliable file system distributed across clusters of machines, while the latter supports a programming paradigm that greatly simplifies sharing work between multiple machines. By combining the two yields a powerful platform for processing “big data” that is endorsed by Internet giants Yahoo, Facebook, Twitter, and many others. This Hadoop-powered extraction process is extremely scalable. Given a cluster of 30 machines, each with two 2.66 GHz processors and 4 GB of RAM, it processes the latest versions of the full English Wikipedia-3.3 million articles, 27 GB of uncompressed markup-in a little over 2.5 hours. The process scales roughly linearly with the size of Wikipedia and the number of machines available (Milne and Witten 2013).

4.4.4 Wikipedia Parsing

In Wikipedia, large number of documents related to the specified concept is organized in the form of graph where group of related pages are linked together. The most important aspect in this approach is that, the topic ontology for a specific concept is developed through extracting the relevant categories in corresponding wikipedia page. Each wikipedia page is explicitly associated with one or more relevant categories. It can provide meaningful information through describing the set of topics related to that concept. Topic ontology is constructed using the relevant categories extracted from the Wikipedia page (Pei et al 2008). In order to facilitate efficient construction of topic ontology, Wiki-parser is newly developed technique to preprocess the content of wikipedia pages that can extract the title, articles, keywords, text, links and relevant categories from wikipedia. Finally, extracted information is
maintained as a text file in metadata warehouse (Klaussner and Zhekova 2011).

4.4.5 Synset Extraction from WordNet

WordNet is used to derive the semantic relation between the extracted relevant categories from wikipedia. There are five steps to extract the semantic relation from WordNet:

- In the beginning, the conceptual category from the available extracted relevant categories is identified using Noun Group parser that can parse the category name linguistically.

- This is followed by extraction of Synset for the selected category from the lexical dictionary (WordNet).

- Next, lexical relation such as Hyponym/Hypernym and Meronym/Holonym from the synset for the selected category has to be determined.

- The semantic relation such as “is a”, “part of” and “related to” is then derived for the list of word pairs according to the determined lexical relation from synset.

- Finally, the lexical and syntactic context using regular expression in large database has to be properly recorded (Girju et al 2006).
4.5 PROPOSED ALGORITHMS IN CONSTRUCTION PROCESS

The corpus based approach relies on two folds i) Concept Acquisition ii) Semantic Relation Extraction algorithm. In first fold, Topic mapping algorithm is newly developed to extract the relevant categories from knowledge specific hierarchically structured semantic wikipedia. As categories in Wikipedia are rich and qualitative topic contents that are available generously. Semantic similarity clustering algorithm computes the semantic similarity measure to group the set of similar concepts. In second fold, Semantic relation Extraction algorithm derives the associated semantic relations between the set of extracted topics from the lexical patterns between synsets in WordNet. Finally, the extracted concepts and their semantic relationship are hierarchically organized in the form of ontology.

4.5.1 Topic Mapping Algorithm

XML /SGML derived standards for information and knowledge structuring and description have become popular in the recent past. These standards share a common goal of simplifying information structuring, accessing and processing by adding structured metadata to information resources. A Topic Map is structure of information that can be used as descriptive metadata for arbitrary data wherein document annotation becomes the most common application. Creation of Topic Maps and their practical applications requires extensive understanding of the knowledge engineering as the domains will have to be intellectually analysed before defining the resource delineation. The proposed topic mapping algorithm, which has significant advantages for knowledge discovery applications due to its low running time and hierarchical clustering capability compared with similar algorithms.
This algorithm can link two available open web resources such as Wikipedia and WordNet to construct topic ontology with near perfect accuracy (Tiun et al 2001). In wikipedia, relevant categories are arranged hierarchically in the form of directed cyclic graph. On the other hand, WordNet has capability to present the well defined taxonomy of synsets. Therefore, a new topic mapping algorithm has been developed to perform mapping between two web resources such as Wikipedia and WordNet. Due to this mapping, set of relevant topics to construct topic ontology has been extracted.

**Pseudo code of algorithm**

The pseudo code of proposed algorithm is illustrated as follows:

```plaintext
Input:
Categories in ODP: C[k] = (C1, C2, ..., Cp);
Synset S[i] = (S1, S2, ..., Sn);
Relevant Categories: RC[i] = (RC1, RC2, ..., RCm);

Output:
Topics (T1, T2, T3,...,Tv)

Algorithm:
Pre = C[k];
Synset = S[i];
Post = RC[j];
for (int i=0; i<n; i++) {
    for (int j=0; j<m; j++) {
        for (int k=0; k<p; k++){
            if (S[i] = "C[k]") {
                T = C[k];
                Return T; // return topic
            }
            else if (S[i] = "RC[j]") {
                T = S[i];
                Return T; // return topic
            }
        }
    }
}
```

**Figure 4.5 Topic Mapping Algorithm**
The function of the algorithm is explained as follows

- The above algorithm takes two inputs such as relevant categories from Wikipedia and synset from WordNet.
- Output of this algorithm is the set of topics to construct topic ontology.
- Similarity between extracted relevant categories and terms in WordNet is determined through topical mapping and terms which matches with the relevant categories is returned as topic to construct topic ontology (Suchanek et al 2007).

### 4.5.2 Semantic Similarity Clustering Algorithm

Many clustering techniques are studied in the past by authors. Clustering is an unsupervised classification technique (Frigui and Krishnapuram, 1999) where several approaches, like graph theoretic approach and K-means algorithm (Selim and Ismail, 1984) have been studied. The quality of graph theoretic approach result is based on quality of the estimation technique for the density gradient (Koontz et al 1995). K-means is an iterative hill-climbing algorithm suffers the sub optimization limitation which depends on the choice of initial clustering distribution. Further, since these clustering algorithms mainly use vector space model (VSM) to represent text, for example, each unique word in vocabulary represents one dimension in vector space. Therefore, these algorithms become insufficient when bag of words representation is adopted because it uses matches produced by keywords. Further, one concept can be portrayed using many different terms. Experiments conducted by using cluster-based approach for semantic similarity in the biomedical domain confirmed the efficiency of this method with similarity measure giving the best overall results of correlation with
human ratings. Fuzzy membership function has been used for measuring Semantic Similarity between any two categories.

• One of the key issues in all fuzzy sets is how to determine fuzzy membership functions.

• The membership function fully defines the fuzzy set.

• A membership function provides a measure of the degree of similarity of an element to a fuzzy set.

• Membership functions can take any form, but there are some common examples that appear in real applications.

• Membership functions can either be chosen by the user arbitrarily, based on the user’s experience (MF chosen by two users could be different depending upon their experiences, perspectives, etc.) or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc.)

• There are different shapes of membership functions; triangular, trapezoidal, piecewise-linear, Gaussian, bell-shaped, etc.

![Figure 4.6 Triangular Membership function](image)

Figure 4.6 Triangular Membership function
Figure 4.6 shows the triangular membership function. In the equation given below a, b and c represent the x coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A (a: lower boundary and c: upper boundary where membership degree is zero, b: the centre where membership degree is 1)

$$\mu_A(x) = \begin{cases} 
0 & \text{if } x \leq a \\
\frac{x-a}{b-a} & \text{if } a \leq x \leq b \\
\frac{c-x}{c-b} & \text{if } b \leq x \leq c \\
0 & \text{if } x \geq c 
\end{cases} \quad (4.1)$$

In the proposed approach, the task is to obtain a score between 0 and 1 inclusively that will indicate the similarity between two topics T1 and T2 at semantic level. The main idea is to find, for each word in the first topic, the most similar matching in the sub topic. The weight of the topic is based on rating the similarity of meaning of the topic pairs on the scale from 0.0 (minimum similarity) to 1.0 (maximum similarity).

The proposed method determines the similarity of two topics from semantic and syntactic information (in terms of common-word order) that they contain. The main advantage of the system is that it has lower time complexity than the other system because the corpus-based measure has been used, while the work combines both corpus-based and WordNet-based measures. The time complexity of the algorithms is given mainly by the number of searches in the corpus and in WordNet. The string similarity measure on short strings, but this is very fast, because the strings are short. The proposed method has been used as unsupervised or supervised. The proposed semantic text similarity method by setting common-word order similarity factor, $wf \in [0, 0.5]$ to observe the effects of common-word
order similarity on both data sets for determining short text similarity. Table 4.2 shows that the proposed Semantic Similarity Clustering Method achieves a higher performance coefficient with the average human similarity ratings.

### Table 4.2 Topic Data Set Results

<table>
<thead>
<tr>
<th>Topics</th>
<th>Bag of Words (Sub-Topic Pairs)</th>
<th>Human Similarity (Mean)</th>
<th>Semantic Similarity Clustering Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cricket</td>
<td>Bat, ball</td>
<td>0.11</td>
<td>0.16</td>
</tr>
<tr>
<td>Personal</td>
<td>Stories, Hobbies</td>
<td>0.11</td>
<td>0.17</td>
</tr>
<tr>
<td>Humor</td>
<td>Jokes, laugh</td>
<td>0.15</td>
<td>0.22</td>
</tr>
<tr>
<td>Marketing</td>
<td>Products, services</td>
<td>0.11</td>
<td>0.16</td>
</tr>
<tr>
<td>Video games</td>
<td>Car race, bounce ball</td>
<td>0.14</td>
<td>0.27</td>
</tr>
<tr>
<td>Thoughts</td>
<td>Ideas, own stories</td>
<td>0.05</td>
<td>0.19</td>
</tr>
<tr>
<td>Sales</td>
<td>Products, services</td>
<td>0.09</td>
<td>0.31</td>
</tr>
<tr>
<td>Sketching</td>
<td>Sketch, water mark</td>
<td>0.02</td>
<td>0.13</td>
</tr>
<tr>
<td>Shuttle cock</td>
<td>Cock, bat</td>
<td>0.17</td>
<td>0.28</td>
</tr>
<tr>
<td>Workshops</td>
<td>Bike, Car</td>
<td>0.16</td>
<td>0.23</td>
</tr>
<tr>
<td>Schools, colleges</td>
<td>Subjects, technical</td>
<td>0.28</td>
<td>0.29</td>
</tr>
</tbody>
</table>

The research work uses a topic directory such as Dmoz (where about 590,000 categories are available) to collect category descriptions. Semantic similarity for pairs of category descriptions has been computed using our proposed method, because they are short texts. This approach would allow evaluation over a large number of pairs of topic descriptions, and we have estimated the expected similarity scores from the positions of the nodes based on weight in the topic ontologies available in Dmoz.
It is an efficient clustering algorithm used to identify the semantically similar topics between the related terms from wikipedia and frequent attributes from WordNet. It is determined from the semantic similarity number of lexical patterns available for that topic in WordNet. Semantic similarity between the topics is computed depending upon the number of lexical patterns applicable between the corresponding topics. If two topics have high semantic similarity score, it represents that more lexical patterns are applicable between them (Bollegala et al 2009). Like that, semantic similarity score for each topic is determined and clusters the topic pairs which have high similarity score with large number of lexical relations pattern relationships. The proposed algorithm can scale down linearly depending upon the number of lexical patterns extracted between the two topics.

**Algorithm: Semantic similarity Clustering Algorithm**

This algorithm performs three different tasks such as

- This algorithm takes three inputs such as related terms, frequent Attributes and WordNet.
- Automatically extracts the lexical patterns for the topic that are needed to make semantic relation to the super class topic.
- Semantic similarity scores for each topic pairs (related terms and frequent terms) is computed according to the number of extracted lexical patterns relationships.
- Topics that have high semantic similarity score are clustered into the super class topic. Like that, semantic similarity score
for all topics are computed with its corresponding super class topic and clusters the topics with high semantic similarity score. The Semantic Similarity Score (SSS) for the two topics (T1&T2) in taxonomy is determined as the logarithmic value of ratio between the weight of sub class topic and super class topic.

\[
SSS = \log \frac{w(T_2)}{w(T_1) + w(T_2)}
\]

(4.2)

Here, \( w(T_1) \) is the weight of super class topic and \( w(T_2) \) is the weight of sub class topic.

If the value of SSS is less than \( \phi \), topic forms the semantic relation to the super class topic. Otherwise, super class discards those topics. Therefore, it may form hierarchy with other topics in the ontology. In information theory, the information content of a class or topic T is measured by the negative log likelihood, where \( \log w(T) \) represents the prior weight that any object is classified under topic T. The semantic similarity between two topics T1 and T2 in taxonomy is measured as the ratio between the meaning of their lowest common ancestor and their individual meanings (Maguitman et al 2010).
Figure 4.7 Semantic Similarity Clustering Algorithm

The semantic similarity clustering method has been used to automatically identify if two topic segments are paraphrases of each other. In the proposed work, Dmoz corpus consisting of 4,076 training and 1,725 test pairs, and determine the number of correctly identified paraphrase pairs in the corpus using the semantic similarity clustering measure. Eleven different similarity thresholds ranging from 0 to 1 with interval 0.1 have been used. For example, using test data set when the similarity threshold 0.6 has used, proposed method predicts 1369 pairs as correct, out of which 1022 pairs are correct among the 1725 manually annotated pairs.
4.5.3 Semantic Relation Extraction Algorithm

After extracting the set of topics to construct topic ontology, the semantic relation between the extracted topics to construct the taxonomy of concepts in the ontological point of view has been determined. To achieve this, lexical patterns between topics are extracted from WordNet. But, the same word pairs can possess different lexical relationship depending upon the various contexts in which two topics are co-occurred as a word pairs. Therefore, lexical patterns between the topics are extracted only for the retrieved contexts which are used to represent the semantic relation between two topics. The main fact is that same semantic relation can be expressed through one or more lexical pattern. Here, the semantic relation such as “is a”, “part of” and “related to” for the extracted topics is determined according to the lexical patterns named as Hyponym/Hypernym and Meronym/Holonym in WordNet.

```
Input:
Lexical Pattern = {Hyponym, Hypernym, Meronym, Holonym}

Output:
Semantic relation = "is a", "part of", "related to"

Algorithm:
If (lexical Pattern == Hyponym)
{
    Semantic relation = "is a",
}
else if (lexical Pattern == Hypernym)
{
    Semantic relation = "is a",
}
else if (Lexical pattern == Meronym)
{
    Semantic relation = "related to",
}
else if (Lexical pattern == Meronym)
{
    Semantic relation = "part of",
}
```

Figure 4.8 Semantic Relation Extraction Algorithm
4.6 ONTOLOGY CREATION

Jena API 2.6.4 is a Java API (Application Programming Interface) framework (Jena 2011) that provides classes and interfaces to construct ontologies using the set of extracted semantic concepts and their corresponding relationships. The constructed ontology is represented in the form of semantic markup language called Web Ontology Language (OWL). Protégé 3.0 is ontology visualization and edition tool with the ezOWL plug-in provides the platform to visualize the automatically constructed topic ontology successfully and also generates the concept class hierarchy with final hierarchy of classes and subclasses depicted in figure 4.9 (Horridge et al 2007). It is also used to perform a set of formal tests to evaluate the completeness and correctness of constructed topic ontology in terms of consistency and redundancy.

Figure 4.9 An Ontology example developed on protégé tool
4.7 ONTOLOGY VISUALIZATION

Protégé is a free, open source ontology editor tool and also considered as a knowledge acquisition system to construct topic ontology. In order to model the ontology, Protégé tool allows two methodologies called frame based and OWL based approach. In this research work, OWL methodology is used to construct topic ontology automatically without any user interface. The foremost feature of the protégé tool relies on two design goals such as i) Ontology visualization: to visualize the constructed ontology in the form of knowledge representation (ontology) ii) Concept Class Hierarchy: it is a straightforward approach that automatically creates and displays the set of concepts or entities in the ontology in the form of class hierarchy with parent and child relationships.

Figure 4.10 Automatic Ontology Creation Process with Protégé
Figure 4.10 illustrates the ontology visualization of the proposed automatic ontology construction creation in a protégé tool. The protégé window depicted in figure 4.10 is divided into two regions as ontology visualization region and concept class hierarchy region. The half part of the window is mostly covered for the ontology visualization region (right side) which is dedicated to visualize the constructed topic ontology. Concept class hierarchy region that lies on left side of the window is dedicated to manage and display the set of topics in the form of parent and child relationships.

4.8 CONSTRUCTED TOPIC ONTOLOGY

In various case studies, it is analyzed and understood that the proposed system can automatically construct the topic ontology using the set of conceptually related topics for the categories in the specific concept.

Figure 4.11 Topic Ontology for the Bank Concept in ODP Directory

In Figure 4.11, Topic ontology is constructed for the category of “Bank” in the existing ontology ODP. The system automatically extracts the
conceptually related terms for the category “Bank” from knowledge based web resources called Wikipedia and WordNet. Therefore, automatically constructed ontology mostly covers the set of topics and also depicted in the knowledge representation using Jena API and Protégé tool. Figure 4.12 shows the topic ontology for the category Card in ODP directory.

![Figure 4.12 Zoom into Topic Ontology for the Category “Card” in ODP Directory](image)

The proposed ontology construction approach is highly automatic and also eliminates the user interaction completely. The main goals of this work are to research and implement an automatic methodology for topic ontology construction with high specific level topics concepts and their corresponding semantic relations. The more optimal number of high quality topics are suggested and extracted through this approach. It provides more support to annotate the web documents in the form of topic ontology with high quality of semantic topics and different kinds of semantic relations are discovered. Figure 4.13, 4.14 and 4.15 shows the Usecase Diagram,
Sequential Diagram and Activity Diagram by using CASE (Computer Aided Software Engineering) tools for ontology visualization.

Figure 4.13 Ontology Visualization using Usecase Diagram
Figure 4.14 Ontology Visualization using Sequence Diagram
Figure 4.15 Ontology Visualization using Activity Diagram
4.9 RESULTS AND DISCUSSION

This section presents the detailed experimental results for classification accuracy of web documents based on the constructed topic ontology. It also describes about the descriptions and characteristics of the data sets used for performing experiments.

4.9.1 Support Vector Machine

The Support Vector Machine (SVM) is a powerful machine learning tool based on firm statistical and mathematical foundations concerning generalization and optimization theory. It offers a robust technique for many aspects of data mining including classification, regression, and outlier detection. SVM was first suggested by Vapnik in the early 1970’s but it began to gain popularity in the 1990’s. SVM is based on Vapnik’s statistical learning theory (Vapnik 1998) and falls at the intersection of kernel methods and maximum margin classifiers. Support vector machines have been successfully applied to many real-world problems such as face detection, intrusion detection, handwriting recognition, information extraction, and others.

- SVM Classification Type

Support Vector Machine is an attractive method due to its high generalization capability and its ability to handle high-dimensional input data. Compared to neural networks or decision tree, SVM does not suffer from the local minima problem, it has fewer learning parameters to select, and it produces stable and reproducible results. If two SVMs are trained on the same data with the same learning parameters, they produce the same results independent of the optimization algorithm they use. However, SVMs suffer from slow training especially with non-linear kernels and with large input
data size. Support vector machines are primarily binary classifiers. Extensions to multi-class problems are most often done by combining several binary machines in order to produce the final multi-classification results. The more difficult problem of training one SVM to classify all classes uses much more complex optimization algorithms and is much slower to train than binary classifiers.

- The SVM Classifier

The Support Vector Machine (SVM) is a well known large margin classifier proposed by (Vapnik 1998). The basic concept behind the SVM classifier is to search an optimal separating hyperplane, which separates two classes. The perfect separation is not often feasible, so slack variables $\xi_i$ can be used which measure the degree of misclassification. Let us consider a classifier whose decision function is given by:

$$f(x) = \text{sign}(x^Tw + b)$$

(4.3)

where $x$ denotes a feature vector and $w$ is a weight vector. Then the SVM algorithm minimizes the objective function:

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

(4.4)

subject to: $y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i > 0, i = 1, 2, \ldots, n.$

This problem leads to so called dual optimization problem and finally (considering non-linear decision hyperplane and using the kernel trick) to:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i,x) + b\right)$$

(4.5)
where $0 \leq a_i \leq C$, $i = 1, 2, \ldots, N$ are nonnegative Lagrange multipliers, $C$ is a cost parameter, that controls the tradeoff between allowing training errors and forcing rigid margins, $x_i$ are the support vectors and $K(x_i, x)$ is the kernel function.

- **Binary Support Vector Classification**

  Binary classification is the task of classifying the members of a given set of objects into two groups on the basis of whether they have some property or not. Many applications take advantage of binary classification tasks, where the answer to some question is either a yes or no. For example, product quality control, automated medical diagnosis, face detection, intrusion detection, or finding matches to a specific class of objects.

  The mathematical foundation of Support Vector Machines and the underlying Vapnik-Chervonenkis dimension (VC Dimension) is described in details in the literature covering the statistical learning theory (Vapnik 1998) and many other sources. In this section we briefly introduce the mathematical background of SVMs in the linearly separable and non-linearly separable cases. One of the attractive properties of support vector machines is the geometric intuition of its principles where one may relate the mathematical interpretation to simpler geometric analogies.

- **Linearly Separable Case**

  In the linearly separable case, there exists one or more hyper planes that may separate the two classes represented by the training data with 100% accuracy. Figure 4.16(a) shows many separating hyper planes (in the case of a two-dimensional input the hyper plane is simply a line). The main question is how to find the *optimal* hyper plane that would maximize the accuracy on the test data. The intuitive solution is to maximize the gap or margin separating
the positive and negative examples in the training data. The optimal hyperplane is then the one that evenly splits the margin between the two classes, as shown in Figure 4.16(b).

![Figure 4.16 SVM Linearly Separable Case](image)

Figure 4.16 SVM Linearly Separable Case

In Figure 4.16(b), the data points that are closest to the separating hyperplane are called *support vectors*. In mathematical terms, the problem is to find \( f(x) = (w^T x_i + b) \) with maximal margin, such that:

- \( w^T x_i + b = 1 \) for data points that are *support vectors*

- \( w^T x_i + b > 1 \) for other data points

Assuming a linearly separable dataset, the task of learning coefficients \( w \) and \( b \) of support vector machine \( f(x) = (w^T x_i + b) \) reduces to solving the following constrained optimization problem:

find \( w \) and \( b \) that minimize:

\[
\frac{1}{2} \|w\|^2
\]

subject to:

\[
y_i (w^T x_i + b) \geq 1, \quad \forall i
\]
Note that minimizing the inverse of the weights vector is equivalent to maximizing $f(x)$.

This optimization problem can be solved by using the Lagrangian function defined as:

$$L(w, b, \alpha) = \frac{1}{2}w^Tw - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + b) - 1], \text{ such that } \alpha_i \geq 0, \forall i$$

where $\alpha_1, \alpha_2, \ldots, \alpha_N$ are Lagrange multipliers and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$.

The support vectors are those data points $x_i$ with $\alpha_i > 0$, i.e., the data points within each class that are the closest to the separation margin.

Solving for the necessary optimization conditions results in

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

where, $\sum_{i=1}^{N} \alpha_i y_i = 0$

By replacing $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ into the Lagrangian function and by using $\sum_{i=1}^{N} \alpha_i y_i = 0$ as a new constraint, the original optimization problem can be rewritten as its equivalent dual problem as follows:

Find $\alpha$ that maximizes

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to $\sum_{i=1}^{N} \alpha_i y_i = 0, \alpha_i \geq 0, \forall i$
The optimization problem is therefore a convex quadratic programming problem which has global minimum. This characteristic is a major advantage of support vector machines as compared to neural networks or decision trees. The optimization problem can be solved in $O(N^3)$ time, where $N$ is the number of input data points.

- **SVM Primal and Dual form**

The research work have seen that for an SVM learning a linear classifier

$$f(x) = w^T x + b$$  \hspace{1cm} (4.6)

is formulated as solving an optimization problem overview:

$$\min_{w \in \mathbb{R}^d} \|w\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f(x_i))$$  \hspace{1cm} (4.7)

This quadratic optimization problem is known as the primal problem.

Instead, the SVM can be formulated to learn a linear classifier

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b$$  \hspace{1cm} (4.8)

by solving an optimization problem over $\alpha_i$.

This is known as the dual problem, and we will look at the advantages of this formulation.

- **Primal and Dual Formulations**

Primal version of classifier:

$$f(x) = w^T x + b$$  \hspace{1cm} (4.9)
Dual version of classifier:

\[
f(x) = \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b
\]  

(4.10)

- **Categorization using SVM**

  Support vector machines are useful for blog classification since thin data and huge dimensionality are dealt with SVM. SVM is to learn and simplify an input output mapping. Inputs are the tags and contents and output is their corresponding topical categories. SVM is used to learn that how to categorize the blogs after the completion of preprocessing and transformations. SVMs have been proven as one of the most powerful learning algorithms for blog classification (Pilaszy, 2005). SVM classifier is used in bioinformatics gene expression and in binary classification (Bayoudh et al 2008). Joachims proposes the topic classification using SVM. In SVM there are two classes involved for categorization task. Two classes are denoted using \((p_1, q_1), ..., (p_n, q_n)\). \(p_i\) denotes the vector feature and \(q_i\) is linearly separable. SVM finds the weight of the vector \(V\) (Kolari et al 2006):

\[
\|V\|^2 \text{ is min}
\]

**Inputs:** tags and contents \((t, c) \Rightarrow p_i \in R(t,c)\), \(b\) is a constant:

\[
V.p + b = 0
\]

A hyper plane \((V, b)\) that separates the blog, this gives the function:

\[
f(p) = \text{sign} (V.p + b), \text{ categorizes the blog}
\]

\[
pi*V+b \geq \pm 1 \text{ when } qi = \pm 1, \text{ canonical hyper plane}
\]

\[
qi (pi*V+b) \geq 1, > = \text{ functional distances}
\]
Normalization of level of vector V for obtaining geometric distance from the hyper plane:

\[ t_c((V, b), \pi) = \frac{q_i (\pi * V = b)}{\|V\|} \geq \frac{1}{\|V\|} \]  \hspace{1cm} (4.11)

Linear combination of vector V is,

\[ L_i (q_i (V. \pi + b) - 1) = 0 \]  \hspace{1cm} (\forall i) \quad c \text{ reduces the margin error. SVM performs on high dimensionality of inputs and it classifies the blogs accurately.}

Support Vector Machines (SVM) is the most popular supervised machine learning algorithm used to classify the web documents into two categories (i.e.) relevant or non relevant. SVM for classification tasks are more effective and competence. It classifies the web documents based on the topic relevance in topic ontology. SVM has been proven as one of the most powerful supervised learning algorithm for classification of web documents. This algorithm takes input as web documents represented in the form of vectors and also deals each documents as a multi dimensional feature gap. It categorizes the documents by taking two inputs as follows:

- Web snippets \( S = \{s_1, s_2, s_3 \ldots \ldots, s_n\} \) where \( n \) number of items are extracted from the web pages returned as results from the web search engines.

- Topic Document Set \( T = \{t_1, t_2, t_3 \ldots \ldots, t_m\} \) consist of \( m \) number of topics in topic ontology.
4.9.2 Classification Accuracy

Experiments are carried out through collecting web snippet data set from the web documents and training set from the set of topics from the constructed topic ontology. Based on the web snippets in test data, corresponding set of topics are collected from the topic ontology. SVM classifier is trained with the training set that aids to categorize the web documents according to the test set of web snippets.

A simple method to achieve the goal of correctly classifying the data in each class is to consider those items with class labels. A new method is to use the binary-label data more than once when training, using each example as a positive example of each of the classes to which it belongs. Table 4.3 shows the experimental datasets with their class labels.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Training Datasets</th>
<th>Testing Datasets</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel</td>
<td>875</td>
<td>88</td>
<td>963</td>
</tr>
<tr>
<td>Sports</td>
<td>910</td>
<td>91</td>
<td>1001</td>
</tr>
<tr>
<td>Entertainment</td>
<td>890</td>
<td>89</td>
<td>979</td>
</tr>
<tr>
<td>Banking</td>
<td>750</td>
<td>75</td>
<td>825</td>
</tr>
<tr>
<td>Arts</td>
<td>450</td>
<td>45</td>
<td>495</td>
</tr>
<tr>
<td>General</td>
<td>624</td>
<td>62</td>
<td>686</td>
</tr>
<tr>
<td>Business</td>
<td>678</td>
<td>68</td>
<td>746</td>
</tr>
<tr>
<td>Engineering</td>
<td>512</td>
<td>51</td>
<td>563</td>
</tr>
<tr>
<td>Science</td>
<td>479</td>
<td>48</td>
<td>527</td>
</tr>
<tr>
<td>Media</td>
<td>787</td>
<td>79</td>
<td>866</td>
</tr>
<tr>
<td>Politics</td>
<td>664</td>
<td>66</td>
<td>730</td>
</tr>
<tr>
<td>Research</td>
<td>224</td>
<td>23</td>
<td>247</td>
</tr>
</tbody>
</table>
Figure 4.17 Classification accuracy between ODP and Topic Ontology

Figure 4.17 shows classification accuracy between ODP and topic ontology based categorization using SVM classifier. The results demonstrate that accuracy of classification was highest when there was less number of web documents for both topic ontology and ODP. In comparison with ODP, topic ontology had significantly higher percentage of classification of accuracy. The performance of both the methods decreased with increasing number of web documents. Yet, the trend of topic ontology based categorization was significantly higher than ODP.

The results demonstrate that topic ontology based categorization produces more accurate classification than existing ODP based categorization. The superior performance of the constructed topic ontology is due to the enrichment of topic ontology with more relevant set of topics. Further, conceptual semantic relation is associated with it. Consequently, topic ontology can classify web documents at a high classification accuracy of 92%. On the contrary, ODP based categorization achieves significantly lower classification accuracy than topic ontology. ODP yields category document
set that only expose the predefined categories related to the concepts arranged in knowledge hierarchy, which explains the lower performance.

4.9.3 SVM Scalability Challenges

The research work discusses the common usability and scalability issues of support vector machines. The SVM scalability challenges have been summerized as follows:

- Optimization requires $O(n^3)$ time and $O(n^2)$ memory for single class training, where $n$ is input size (depends on algorithm used). To address this issue, new optimization algorithms continue to be introduced.

- Multi-class training time is much higher, especially for optimization.

- Multi-class performance depends on approach used and deteriorates with more classes.

- SVM is impractical for large input datasets, especially with non-linear kernel functions.

4.9.4 Performance Evaluation

Performance of the constructed topic ontology is measured based on standard metrics, such as precision, recall and F-measure. Therefore, experiments were conducted to evaluate the variation in the number of ODP concepts matched against the extracted terms with respect to the string metric chosen (Table 4.4). The prototype system used precision and the recall of the similarity metrics.
Table 4.4 String Similarity Metrics Comparison

<table>
<thead>
<tr>
<th>String Similarity Metrics</th>
<th>Concepts in ODP</th>
<th>Matched Concepts</th>
<th>Effective Number of Concepts Matched</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF</td>
<td>29</td>
<td>15</td>
<td>7</td>
<td>46.66</td>
<td>51.72</td>
</tr>
<tr>
<td>Jaro Winkler</td>
<td>29</td>
<td>29</td>
<td>14</td>
<td>48.27</td>
<td>100.00</td>
</tr>
<tr>
<td>Jaro Winkler TFIDF</td>
<td>29</td>
<td>17</td>
<td>13</td>
<td>76.47</td>
<td>58.62</td>
</tr>
<tr>
<td>Jaccard</td>
<td>29</td>
<td>3</td>
<td>3</td>
<td>100.00</td>
<td>10.34</td>
</tr>
<tr>
<td>Level2Jaro</td>
<td>29</td>
<td>29</td>
<td>17</td>
<td>58.62</td>
<td>100.00</td>
</tr>
<tr>
<td>Level2Jaro Winkler</td>
<td>29</td>
<td>29</td>
<td>17</td>
<td>58.62</td>
<td>100.00</td>
</tr>
</tbody>
</table>

**Precision**

In Information Retrieval (IR), precision is the portion of retrieved instances that are relevant and a measure of the proportion of selected items that the system got right.

\[
\text{Precision} = \frac{|\text{Correctly Matched Concepts}|}{|\text{Total Number of Matched Concepts}|} \quad (4.12)
\]

**Recall**

In IR, recall is the portion of relevant instances that are retrieved and a measure of the proportion of the target items that the system selected.

\[
\text{Recall} = \frac{|\text{Correctly Matched Concepts}|}{|\text{Total Number of Concepts that Should Matched}|} \quad (4.13)
\]
F-Measure

F-measure combines recall and precision into one measure with equal weights and is defined as

\[
F\text{- Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\] (4.14)

4.9.5 Evaluation and Comparison

For the experimental evaluation, the proposed work used

- A corpus containing a collection of 15 texts from the software development literature available at the Wikipedia, an online encyclopedia. From this corpus a set of 450 terms and 19 associations were extracted by using the semantic similarity extraction algorithms.

- A pattern catalogue composed of two ontology design patterns, which had 29 concepts and 28 associations.

- A reference list of 98 matches, created from the text corpus and the ODP concepts for find the correct matches. This reference list included the exact matches, in addition to the terms that are not exact matches but have the same meaning as the concepts in the ODPs.

Figure 4.18, 4.19 and 4.20 illustrates the number of concepts matched, precision and recall value obtained using the algorithms. In the case of matching the number of concepts, it was found that the matched concepts were near perfect in the three of the concepts used. The effective matching of the concepts were up to 50% (Figure 4.18). In the case of precision measure, Jaccard had the highest precision Overall, the precision ranged from 45-60%,
with maximum at 100% for Jaccard (Figure 4.19). At the same time, recall measures showed variation of 45% to 100%. Jaccard value touching as low as 10% (Figure 4.20).

A comparison of the string metrics for the string similarity metrics, number of matches, precision and recall demonstrated that precision was highest for Jaccard (100%), followed by Jaro Winkler (78%), while others were in the range of 45% to 60%. Recall was the best at 100% for Jaro Winkler, Level2Jaro and Level2Jaro Winkler. On the contrary, Jaccard had the least recall (10%). Except for Jaccard, all others had F-measure values between 45% and 70% indicating a combination of precision and recall produced better results (Figure 4.21).

According to the different precision, recall and F-Measure values obtained by the algorithms, we can deduce that the reliability of the concepts and associations generated by the ontology is closely linked to the string similarity metric used for the terms and concepts matching process. As a result, the choice of a string similarity metric with a high precision value will imply the construction of a topic ontology having few ODP concepts.

![String Similarity Metrics Comparison](image)

**Figure 4.18 Number of Concepts Matched with the String Similarity Metrics**
Figure 4.19 Precision Value for Similarity Metrics

Figure 4.20 Recall Value for Similarity Metrics

Figure 4.21 Comparison of Different String Similarity Metrics
4.10 CONCLUSION

In this research work, a novel approach for automatic construction of topic ontology has been proposed. The proposed approach is more automatic and completely eliminates the user intervention. The main goal achieved is the implementation of topic ontology construction with specific level of topic concepts and their corresponding semantic relations. The more optimal number of high quality topics are suggested and extracted through this approach. It provides more support to annotate the web documents in the form of topic ontology with a high quality of semantic topics and different kinds of semantic relations are discovered. A software prototype was created and implemented to support the topic ontology construction process. The proposed approach was based on concept acquisition and semantic relation extraction. A novel topic mapping algorithm was proposed to acquire the concepts related to the category in ODP. The semantic similarity clustering algorithm was proposed to group the identified concepts to determine the semantic distance between the concepts. In the semantic relation extraction algorithm, the semantic relation between the identified concepts is derived from the lexical relations of the WordNet. The Jena API framework hierarchically organized the identified concepts and their semantic relationship into topic ontology. The protégé tool is effectively deployed to visualize the constructed topic ontology. Finally, the performance of the constructed topic ontology is better than ODP in terms of classification accuracy.