Chapter 5

CLASSIFICATION ALGORITHMS FOR METADATA EXTRACTION

5.1 INTRODUCTION

The metadata stored in the learning object repository is provided with specific tagging in the semantic web. This stored data is retrieved from the domain ontology based on the specific indexing using the terms which has been extended in the IEEE LOM (Alkhasawneh et al 2008). To obtain these contents different classification algorithms are used to retrieve these data (Bose 2006). The algorithms used for identifying the concept and significance is one such classification algorithm where the data retrieved is evaluated based on the accuracy in terms of precision and recall.

The weighted algorithm is used for text clustering. The words in the text are initially preprocessed by using these algorithms and are further stored in the ontology which are later indexed using the hybrid algorithm discussed in chapter 6. The content similarity algorithm is used to specify the contents with similarities. The performance of these algorithms is discussed and the experimental analysis is carried out to identify its accuracy (Convetini et al 2006). The results given have showed better performance compared to the manual extraction of the metadata which has been stored. To outline these algorithms a set of data has been used for efficient retrieval of the data.

5.2 ALGORITHM FOR CONCEPT AND SIGNIFICANCE IDENTIFICATION

Chia et al (2006) have provided a major survey on web data extraction approaches and have compared them with three dimensions which are the task domain, the automation degree, and the techniques used. The important aspect of the first dimension explains why an IR system fails to handle some web sites of particular structures. The second dimension classifies the system based on the various techniques used. The third criteria measure the degree of automation for the systems (Eduardoj et al 2014). Information extractions across web sites become more important as we move towards semantic web. The survey focuses on data extraction from web documents The page, fetching support and extracting data integrates the document in terms of schema mapping from various data sources.

The classification algorithm is identified for the retrieving the contents based on concept and significance of the query given for searching the document (Maurer and Sapper 2001) and (Romero et al 2008). Figure 5.1 gives the outline of the various approaches used involving the different algorithms for retrieval of documents which exist in the form of learning objects (Alsultanny 2006) and (Gresty and Edwards 2012). Even before the retrieval, the documents are preprocessed (Rosmalen et al 2006) and (Edelson 2001) and then stored where they are retrieved with topic specific indexing algorithm. The preprocessing of documents is done by using the text clustering which used the word net and also the porter stemming algorithm. For the documents related to the domain ontology a preprocessing document analysis and tagging is carried out as depicted in figure 5.1. This preprocessing is carried out for the different document types based on the query given by the user. The semantic search layer focuses on text clustering and document indexing along with the combined approach which uses semantic indexing which is discussed in chapter 6.The pre-processed documents are stored in the repository which focuses on the domain ontology. Further using these approaches the documents are retrieved according to the user query. The figure 5.1 outlines all these approaches related to document retrieval in the form of different layers.

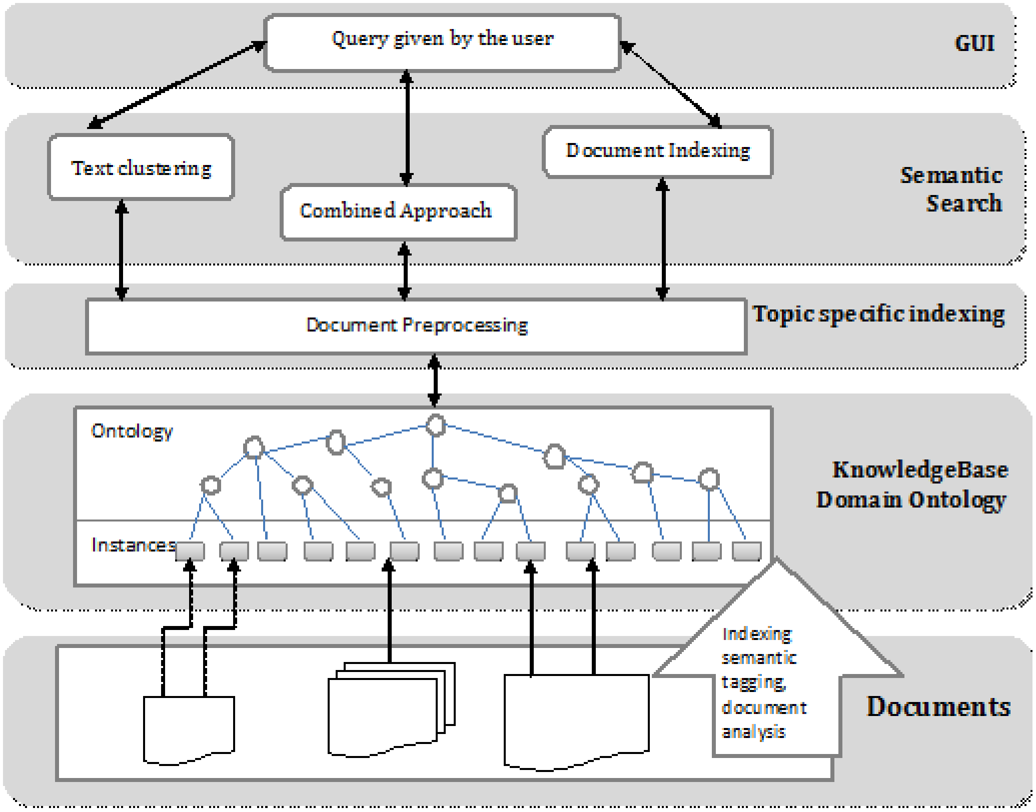


Figure 5.1 Document Retrieval Approaches

According to Brut and Dumitrescu (2011) and Jovanovic et al (2006), the simplest weight is the Term Frequency TF and $TF(t_i, d_c)$ is the number of times term t appears in the document d. The document identified by the user after the execution of the search query is analyzed from the domain storage repository. The identified document is parsed and the tokens (splitting sentences into words) identified are weighed using stop word removal and stemming methods which is discussed in the weighted algorithm for text clustering(Magoulas et al 2003) and(Metallidoua and Platsidoub 2008). The term weight calculation is done using term frequency TF. $TF(t_i, d_c)$ is the number of times the term t occur in the document d_c. The popular method of calculation of weight is given as TFxIDF weighting, where the weight is calculated to be proportional to the frequency of the corresponding term t_i in the document d_c and inversely proportional to the number of documents$|D|$. Using the equation $w_i = TF(t_i, d_c)\log\frac{|D|}{DT\ (t_i)}$ the values are assumed where $DT(ti)$ is the number of documents in the collection D which has the terms t_i. The tokens in the document are identified by parsing techniques and stemming methods to compare the words. The similarity of the vectors formed is obtained by the cosine similarity $\mathrm{sim}(d, q) = \cos(d, q) = d \bullet q / |d||q|$ which is equal to the cosine of the angle formed by the two vectors d and q in the n dimension vector space.

The terms identified may have multiple meanings and to retrieve a document term from a document belonging to a particular domain, we annotate them with a list of concepts. The significance of a concept is identified if more number of related concepts of the particular term occurs in the document (Leutner and Plass 1998) and (Bunderson and Martinez 2000). The proposed classification algorithm is used to retrieve the concept based on the term along with the significance of the term.

For every term identified from the document D the concept C_{ij} is obtained and the significance is also computed C_{ij}Sig. The C_{ij}Sig is taken as the normalized frequency term. For every concept C_{ij} the related concept rc in the document is identified. The associated concept is then normalized by the term frequency to the corresponding terms t and the related concept r_c. Significance$C_{ij} = t_i\,\mathrm{freq} + \alpha * \mathrm{tp}\,\mathrm{freq}$, where we assume α value to be ½, where $t_p = \mathrm{term}$ corresponding to related concept r_c. For the particular term we identify the concept with a maximum significance value. Following are the steps involved in the proposed algorithm to get the input of terms in the document D and to give the output of concepts with their significance.

First we normalize the frequency of the terms and then match the term frequency to the concept significance. Then we find the related concept r_c and its occurrence in the document D by using Significance $C_{ij} = t_i\,\mathrm{freq} + \alpha * \mathrm{tp}\,\mathrm{freq}$ where we assume α value to be ½.For a set of documents D, the relevant documents based on topics identification is given as |R| and the filtered documents based on domain specific concepts is given as |A|. The intersection of |R| and |A |is given as $|R_a|$.

Precision is the fraction of the retrieved documents, which is relevant. $\mathrm{Precision} = \frac{|R_a|}{|A|}$.

Recall is the fraction of the relevant documents, which has been retrieved. Recall $= \frac{|R_a|}{|R|}$.

Finally we select the final concept by identifying its significance with a highest significance score which is to be greater than the threshold value. The algorithm returns the list of the related concepts and the significant value. The combination of both these methods is used for filtering and extracting the query based on the domain ontology.

5.3 CONTENT SIMILARITY INDEXING

The contents are indexed based on the contents stored in the domain ontology and the extended attributes of the IEEE LOM which provides the metadata tagging to the learning object repository (Schiaffino et al 2008) and (Nasullah Khalid et al 2010). According to Veronique (2006) and (Yen et al 2012) the similarity of the content which is present in the repository the document is retrieved using the content similarity algorithm. This algorithm focuses on the learner's choice of topic. According to Yuan et al (2014) and (Burasakorn and Vilas 2012) the domain ontology the topics are identified with a relation to the prerequisite of the topic and the specific terms are retrieved for the given topic.

YaGao et al (2010) and Chou et al (2012) have proposed an innovative method based on data extraction based on index path in web. In this innovative approach they have established the index path for each text node identified using the metadata standards. This technique defines the prefixing of data using data-rich keyword in the indexing path and thereby extracting and obtaining a wrapper accordingly. This method is efficient in the recall and the precision of data extraction but has limited keyword matching in special pages such that it will concentrate on those cases which are prone to be restricted by keyword in the future extraction techniques.

The outline of the algorithm for the content similarity for a learner's choice is given below. The learners are diverse and the personalization of learning contents for these learners can be attained using these algorithms which we have proposed and compared to the existing manual retrieval technique. The learners are classified according to the difficulty level in learning. The system provides a means to give the feedback on the identified learning object. The similarity index for each learner is identified using the similarity cosine of the learning object which is later retrieved and the contents retrieved according to the similar learners are also identified for the similar set of learning objects where SimLO denotes the Similar Learning Objects

The outline of Content Similarity Algorithm

```
Content Similarity Algorithm
Input:The query input from user based on the learner’s choice of topic
Output: The learning object based on the similarity of the content.
Step1 : Initialize the values of the identified similar learners{
Step2 :Return the feedback based on the relevance of the Learning Object LO
Step 3 : Identification of similarity index for the number of learners
Step 4: for each learner identified l in the Sim LO
Step 5: Calculate the value of the similarity cosine value
Step 6: if {(sim index) is equal to or greater than  the threshold value
Step7: assign (l, sim cosine) to SimLO;}
Step8: return the set SimLO;
```

The similarity of the contents is identified according to the learner’s choice of query and the similarity is measured with the sample code for the set of learning objects present in the ontology. The representation of the algorithm for the content similarity indexing algorithm is given the query is given according to the choice of similar content based on the learner’s perception. It also gives the content similarity of the learning objects retrieved. The algorithm uses such techniques for determining the similarity scores of the identified learning objects.

The proposed algorithm is a unique hybrid algorithm which compares the similarity indexing based on latent semantic indexing along with topic indexing which gives better personalization in terms of precision and recall compared to existing algorithms as show in Figure 5.2.

```
        sim1=((productArray2[0]*d1[0])+(productArray2[1]*d1[1]))/(sq1
*sqa1);
        //sim1=Math.round(sim1 * 100);
                // sim1=sim1/100;
        System.out.println("sim(q,d1)"+sim1);
        sim2=((productArray2[0]*d2[0])+(productArray2[1]*d2[1]))/(sq1
*sqb1);
        //sim2=Math.round(sim2 * 100);
                // sim2=sim2/100;
        System.out.println("sim(q,d2)"+sim2);
        Continue for the set of LO’s in the domain ontology.
```

Table 5.1 Sample Algorithm Representation

5.3.1 EXPERIMENTS AND PERFORMANCE OF CLASSIFICATION ALGORITHM

The above said techniques as well as evaluation methods were used for a set of documents. The documents were stored locally and the list of documents for a specific domain is stored. Using the domain ontology and the annotation to the list of documents the topics are filtered without concept identification. Later the documents use the specific indexing methods to filter the set of documents with the concept identification and are retrieved. The domain ontology for a set of documents locally stored is retrieved according to concept and the significance score is also calculated. The precision percentage is obtained by filtering terms with concept identification and without concept identification and also the recall is identified. The graph reflects the better precision and recall for a set of documents in a specific domain for the documents where the related terms were retrieved. The effective personalization is attained in terms precision and accuracy for a specific term with concept identification and significance from the domain ontology. The improvement in precision with concept identification and significance is given in the graph Figure 5.2. The precision percentage is shown along the Y axis and the query search in a set of documents of a specific domain for a specific concept is shown in the X axis. The results have proved that better precision and recall is obtained by concept identification and significance score using the domain ontology and annotation of learning objects by the specific indexing method in Figure 5.2.

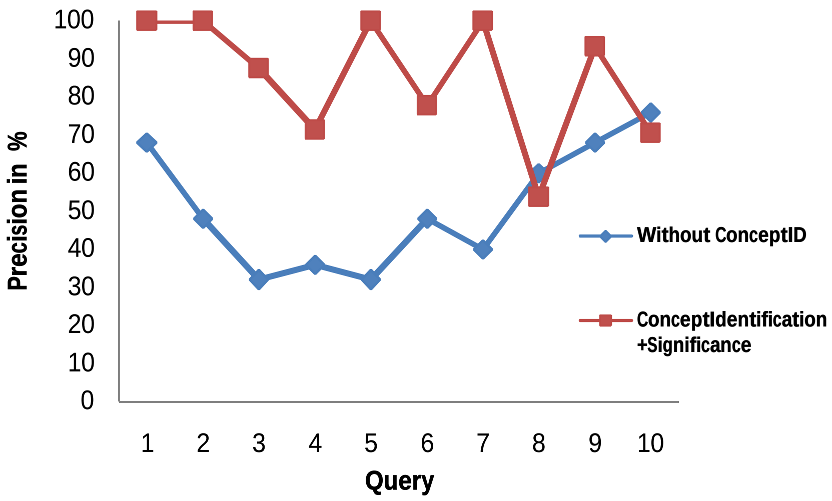


Figure 5.2 Performance of Classification Algorithm

The figure 5.2 describes the performance of the classification algorithm with the technique of using concept along with the significance of the topic. The algorithm performance has been evaluated with the content stored in the local repository as well the ACM classification topics using google search engine.

These attributes have been extended from the basic IEEE LOM attributes. The table 5.2 gives the list of documents retrieved based on the query search of the user. In the table we have considered a sample set of query in which the number of documents has been identified with and without using the classification algorithm. Based on the retrieved content the performance is identified by finding the precision and recall of the documents is identified based on the search query. The significance score is obtained by identifying the similar learning objects based on the content similarity algorithm. From a set of locally stored documents a sample set of ten topics have been given to show the performance of the algorithm and the threshold value depends on the number of documents used to get the maximum number of similar topics and is scalable.

No	Search Query	Without Concept identification& significance	Precision %	Filtered Documents	Filtered with Concept Identification	Precision %using concept identification& Significance	Recall % using concept identification& significance
1	Light	17	68	12	12	100	70.59
2	Wave	12	48	12	12	100	100
3	Wave Length	8	32	8	7	87.50	100
4	Interference	9	36	7	5	71.43	77.78
5	Intensity	8	32	4	4	100	50.00
6	Scattering	12	48	9	7	77.8	75.00
7	Atom	10	40	5	5	100	50
8	Electron	15	60	13	7	53.85	86.67
9	Motion	17	68	15	14	93.33	88.24
10	Reflection	19	76	17	12	70.59	89.47

Table 5.2.Query Search Based on Domain Ontology

5.4 WEIGHTED ALGORITHM FOR TEXT CLUSTERING

The text classification of documents illustrates the dimensionality of the feature vector which is usually very huge (Huynh et al 2005). For example, 20 Newsgroups and Reuters21578 top-10, which are two real-world datasets, both have more than 15,000 features. Such high dimensionality can be a severe obstacle for classification algorithms (Roy et al 2005). To alleviate this difficulty, feature reduction approaches are applied before document classification tasks are performed (Felder and Silverman 1998). Two major approaches, such as feature selection and feature extraction have been proposed for feature reduction. In general, feature extraction approaches are more effective than feature selection techniques, but are more computationally expensive (Kappe et al 2009) and (Bing et al 2012). Therefore, developing scalable and efficient feature extraction algorithms is highly demanded for dealing with high-dimensional document data sets (Ghauth and Abdullah 2010). Classical feature extraction methods aim to convert the representation of the original high-dimensional data set into a lower-dimensional data set by a projecting process through algebraic transformations. The idea of feature clustering is to group the original features into clusters with a high degree of pair wise semantic relatedness. Each cluster is treated as a single new feature, and, thus, feature dimensionality can be drastically reduced to identify the relevant terms (Lo and Shu 2005) and (Meo et al 2007). Matching is the technique used in order to relate or match the various set of related documents. There are various matching techniques present but they are retrieving data which is time consuming for a large number of data. There are various steps used in order to indicate the matching techniques, the graph is being extracted into data models like trees and they are being used in order to represent the various set of data, they are being separated into words and that words are related or matched with the Word Net database and the resultant graph is being produced.

The weighted clustering algorithm is numerical, unsupervised, time-series data stream clustering and iterative clustering algorithm which uses distance method for clustering(Popescu 2010). The distance is the ordinary distance between two points that one would measure with a ruler which in this algorithm is used for calculating distance between the centres and the documents.

Clustering differs from multidimensional scaling (perceptual maps), which aims to depict all the evaluated objects in a way that minimizes the topographical identification by using a few dimensions as possible. Clustering algorithms partition data into a certain number of clusters like subsets, groups and categories (Stefanowski 1998) and (Bing et al 2012).

As illustrated in figure 5.1 the document is to be pre processed with semantic tagging before storing in the domain ontology. For the documents to be pre processed we provide the metadata tagging according to the properties of the IEEE LOM .The text clustering is used to provide certain pre processing. Even before the documents are retrieved with the topic specific indexing algorithm which is discussed in chapter 6 the documents are stored with the weighted clustering algorithm (Snae and Brueckner 2006) and (Tsumoto 2004). TDC using benchmark time series, motion trajectory, and time-series data stream clustering is a document clustering system which clusters the set of documents into two groups based on the user typed key term. Traditionally, clustering techniques do not consider the semantic relationships between words, such as synonymy and hypernym. To exploit semantic relationships, ontology such as Word Net have been used to improve These related word frequencies are also calculated. Then those frequencies are weighted by using the inverted document frequency method.

Text document clustering can greatly simplify browsing large collections of documents by reorganizing them into smaller number of manageable clusters. Existing algorithms have the constraints like grouping but cannot overcome the ambiguity as well as the exact synonyms (Xin and Barbara 2010) and also existing text clustering uses the frequent word sets to cluster the documents.

As given by Ahn (2007) XML data is dynamic in practical application. A novel algorithm called weighted cosine measure (WCM) has improved from the traditional algorithm is proposed, and by using this algorithm it is possible to calculate the similarity between two clusters. XML documents can be effectively clustered and the results of using the WCM are better than using the traditional cosine measure. Then, XML documents in each cluster have similar structures which not often changed.

Text clustering is done by only relating documents that use identical terminology. The bag of words representation is used for these clustering methods. Text document clustering plays an important role in providing intuitive navigation and browsing mechanisms by organizing large amounts of information into a small number of meaningful clusters (Bot et al 2004). Desired results for a variety of temporal data clustering tasks are weighted and cluster ensemble algorithm can combine any input partitions to generate a clustering ensemble. So far, however, existing text clustering solutions (Cabelle et al 2014) only relate documents that use identical terminology, while they ignore conceptual similarity of terms such as defined in terminological resources like “Word Net”. For the preprocessing the documents in text clustering we integrate the conceptual account of terms found in Word Net which is nothing but a lexical database for grouping English words into a set of synonyms called syn sets(Felder and Brent 2005).The weighted algorithms has a has a frequent concept to cluster the text documents. It also uses the algorithm to utilize the semantic relationship between words to create concepts. The set of techniques used for text preprocessing are discussed as follows.

Stop Word removal: Stop words are words which are filtered out prior to, or after, processing of natural language data which is in the text format. It is controlled by human input and not automated (Gabor et al 2012). These are some of the most common, short function words, such as *the*, *is*, *at*, *which* and *on*.

Porter stemming: Stemmers employ a lookup table which contains relations between root forms and inflected forms. To stem a word, the table is queried to find a matching inflection. If a matching inflection is found, the associated root form is returned. Using the stemming algorithm in the example the words "matching", "matched", "match", and "matcher" to the root word, "match" are reduced

Text Classification: Words ending in nym’s are often used to describe different classes of words, and the relationships between words like hypernym (a word with a generalized meaning), hyponym (a word with a specific meaning) and synonym (words with the same meaning). The text analysis is carried out using the ontology using the word net (Kabassi and Virvou 2004). The featured results produced by the sentence-based, document-based, corpus-based, and the combined approach concept analysis have higher quality than those produced by a single-term analysis similarity. Weight clustering (WC) algorithm, which is an incremental feature clustering approach to reduce the number of features for the text classification accuracy. The following indexing techniques are used for the text clustering.

The indexing is done by detecting the combination of distinct weights of document terms automatically (Liber et al 2004). The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster. There are three ways of weighting, hard, soft, and mixed Modified Huber's Index (MHI) defines the proximity matrix of objects and "N" is a cluster distance matrix derived from the partition and the validity index (Litzinger et al 2007) is used to determine the dissimilarity metric between clusters c_1,c_2,c_3…c_n. The technique suggested for indexing uses the Normalized Mutual Information (NMI) in our proposed method which is used to measure the consistency between two partitions that is, the amount of information shared between two partitions.

Feature Document Cluster: Patterns are considered one by one. The user does not need to have any idea about the number of clusters in advance (Mak and Manakata 2002). No clusters exist at the beginning, and clusters can be created if necessary. For each word pattern, the similarity of this word pattern to each existing cluster is calculated to decide whether it is combined into an existing cluster or a new cluster is created. Once a new cluster is created, the corresponding membership function should be initialized. The normalized mutual indexing algorithm is used to retrieve data units which are clustered into different groups with similar semantics. Based on data content, presentation style based on font weight,style,color and text decorator, data style, and adjacency based on preceding and succeeding data the key word containing the data units is retrieved from the text node. The outline of the normalized mutual indexing in weighted clustering algorithm is given as follows

Normalized Mutual Indexing in weighted clustering algorithm.

Input : Key words

Output : Indexed key word based on content, style, presentation and adjacency.

Step 1: The entered input belongs to the data clustering the clustering distribution P_c in c where c is c= {c1,c2,..cn}.

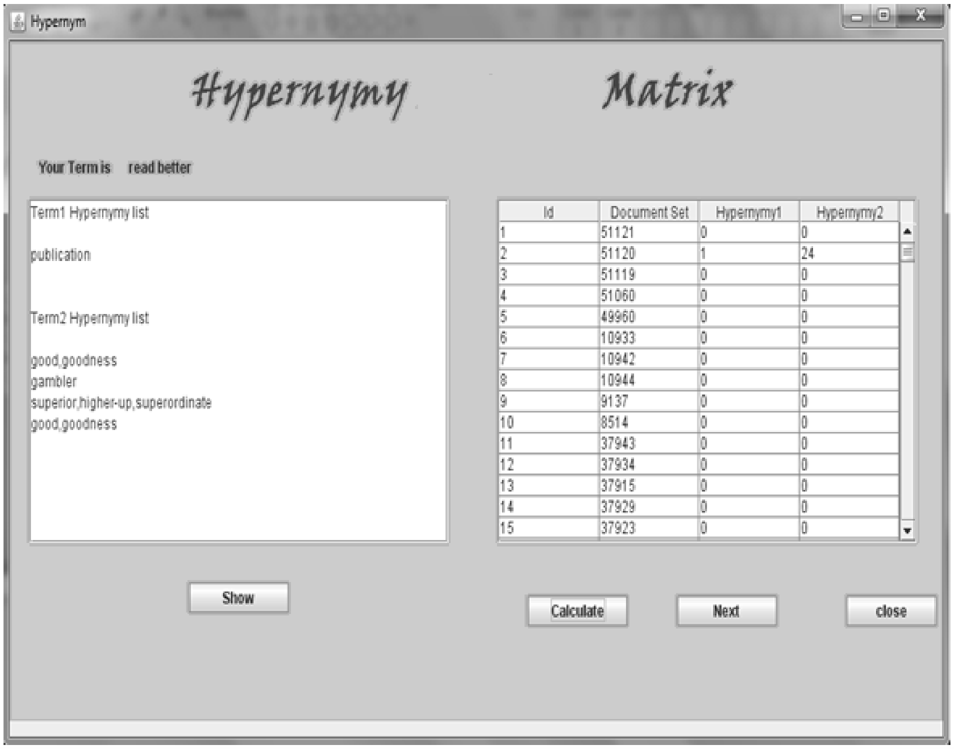
Step 2: Enter the number of points n_i in the ith cluster Ci and N is the cumulative number of points in the ith cluster and P_c (i)= n_i N.

Step 3: While NMI for any distribution P_{c1}=(p1,…,pn)and P_{c2}=(q1,….qn), the mutual index is $I(p, q) = \Sigma\, i, jR(i, j)\, log\, R(i\ , j)\, PC1(i)PC2(j)$ *where* $R(i, j)$ *is the probability distribution*

Step 4: If indexed element in data content is equal to same word, presentation style, adjacency and data content in the cluster.

Step 5: Then retrieve key word indexed based weighted clustering.

Figure 5.3 shows the set of data used in clustering algorithm and also the matrix obtained for the terms retrieved from the database.



Your Term is read better

Term1 Hypernymy list

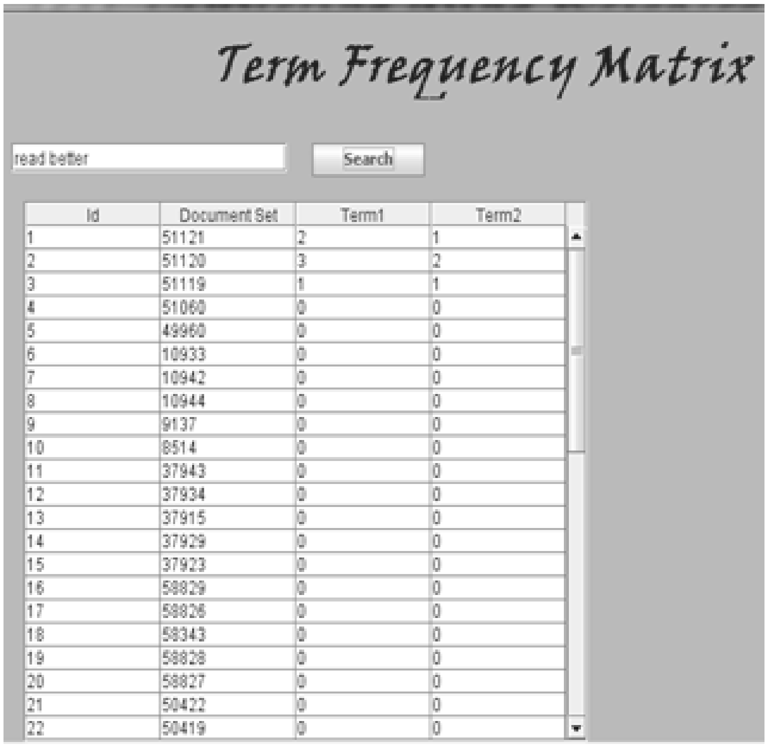
publication

Term2 Hypernymy list

good,goodness
gambler
superior,higher-up,superordinate
good,goodness

Id	Document Set	Hypernymy1	Hypernymy2
1	51121	0	0
2	51120	1	24
3	51119	0	0
4	51060	0	0
5	49960	0	0
6	10933	0	0
7	10942	0	0
8	10944	0	0
9	9137	0	0
10	8514	0	0
11	37943	0	0
12	37934	0	0
13	37915	0	0
14	37929	0	0
15	37923	0	0

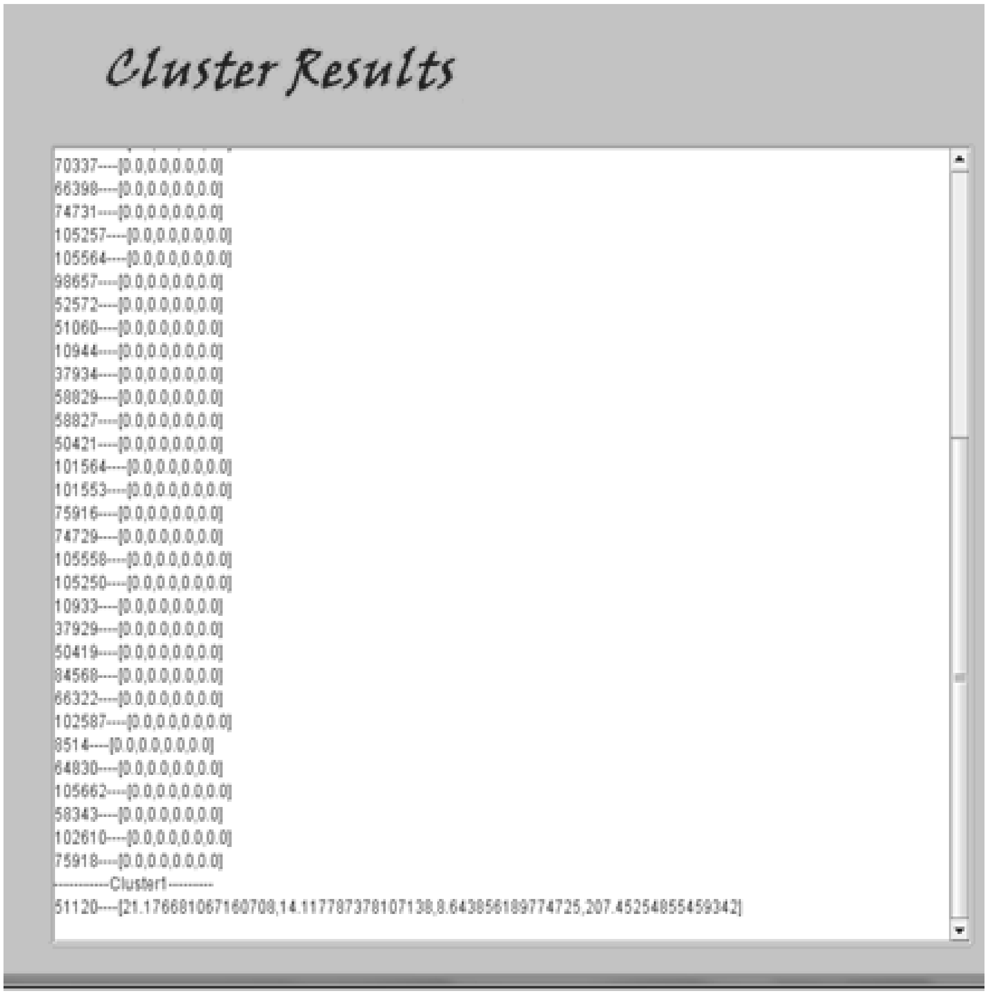
Figure 5.3 Hypernym matrix



Id	Document Set	Term1	Term2
1	51121	2	1
2	51120	3	2
3	51119	1	1
4	51060	0	0
5	49960	0	0
6	10933	0	0
7	10942	0	0
8	10944	0	0
9	9137	0	0
10	8514	0	0
11	37943	0	0
12	37934	0	0
13	37915	0	0
14	37929	0	0
15	37923	0	0
16	58829	0	0
17	58826	0	0
18	58343	0	0
19	58828	0	0
20	58827	0	0
21	50422	0	0
22	50419	0	0

Figure 5.4 Term Frequency Matrix using Weighted Algorithm

The result of the weighted algorithm is given as below. Figure 5.4 gives the term Frequency Matrix using Weighted Algorithm. The values related to the word obtained after clustering results with the document set and the terms related to it. The obtained terms are illustrated in the term frequency matrix. The results have great impact in retrieval of relevant word after the preprocessing which makes the user find the targeted word. The cluster of words is formed and the term matrix is obtained. Further the results of the matrix based on the weighted algorithm is shown in figure 5.5 which displays the cluster results based on the weighted algorithm using normalized mutual indexing.



```
70337----[0.0,0.0,0.0,0.0]
66398----[0.0,0.0,0.0,0.0]
74731----[0.0,0.0,0.0,0.0]
105257----[0.0,0.0,0.0,0.0]
105564----[0.0,0.0,0.0,0.0]
98657----[0.0,0.0,0.0,0.0]
52572----[0.0,0.0,0.0,0.0]
51060----[0.0,0.0,0.0,0.0]
10944----[0.0,0.0,0.0,0.0]
37934----[0.0,0.0,0.0,0.0]
58829----[0.0,0.0,0.0,0.0]
58827----[0.0,0.0,0.0,0.0]
50421----[0.0,0.0,0.0,0.0]
101564----[0.0,0.0,0.0,0.0]
101553----[0.0,0.0,0.0,0.0]
75916----[0.0,0.0,0.0,0.0]
74729----[0.0,0.0,0.0,0.0]
105558----[0.0,0.0,0.0,0.0]
105250----[0.0,0.0,0.0,0.0]
10933----[0.0,0.0,0.0,0.0]
37929----[0.0,0.0,0.0,0.0]
50419----[0.0,0.0,0.0,0.0]
84568----[0.0,0.0,0.0,0.0]
66322----[0.0,0.0,0.0,0.0]
102587----[0.0,0.0,0.0,0.0]
8514----[0.0,0.0,0.0,0.0]
64830----[0.0,0.0,0.0,0.0]
105662----[0.0,0.0,0.0,0.0]
58343----[0.0,0.0,0.0,0.0]
102610----[0.0,0.0,0.0,0.0]
75918----[0.0,0.0,0.0,0.0]
-----------Cluster1---------
51120----[21.176681067160708,14.117787378107138,8.643856189774725,207.45254855459342]
```

Figure 5.5 Cluster Results based on Weighted Algorithm

5.5. SUMMARY

In this chapter we have discussed the classification algorithm for concept and significance. We have also used the weighted algorithm for text clustering by which the pre-processing is done and retrieved by the various semantic indexing algorithms which is discussed in chapter six.