International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Wordnet Based Document Clustering

Madhavi Katamaneni[1], Ashok Cheerala [2]

[1]Assistant Professor VR Siddhartha Engineering College, Kanuru, Vijayawada, A.P., India

[2]M.Tech, VR Siddhartha Engineering College, Kanuru, Vijayawada, A.P, India

Abstract: ***Document clustering is considered as an important tool in the fast developing information explosion era. It is the process of grouping text documents into category groups and has found applications in various domains like information retrieval, web information systems. Ontology based computing is emerging as a natural evolution of existing technologies to design with the information onslaught. In current dissertation work, background knowledge derived from WordNet as ontology is applied during preprocessing of documents for document clustering. Document vectors constructed from WordNet synsets is used as input for clustering. Comparative analysis is done between clustering using k-means and clustering using bi-secting k-means. A document Categorization tool is developed which summarizes the hierarchy of concepts obtained from WordNet during clustering phase. GUI tool contains the association between WordNet concepts and documents belonging to the concept.***

Keywords: Document clustering, Ontology, BOW, POS Tagging, Stemming, Labeling

1. Introduction

With the abundance of text documents available through the Web and corporate document management systems, the partitioning of document sets into previously unseen categories ranks high on the priority list for many applications like business intelligence systems. Nowadays the problem is often not to access text information but to select the relevant documents.

The steady development of computer hardware technology in the last few years has led to large supplies of powerful and affordable computers, data collection equipments, and storage media. These technologies provide good support to the database and information industry and make a huge number of databases and information repositories available for transaction management, information retrieval, and data analysis. Therefore these technologies provide a huge volume of the text documents available on the Internet, digital libraries, news sources and company-wide intranets. With the increase in the number of electronic documents, it is hard to manually organize, analyze and present these documents efficiently.

a) Motivation

By grouping similar sets of information, an organized document structure can be formed, which reduce the search space and help users to access a number of related documents. Document organization can be done by document classification and document clustering. Data mining is the process of extracting the implicit, previously unknown and potentially useful information from data. Clustering or segmentation of data is a fundamental data analysis step that has been widely studied across multiple disciplines for over 40 years. Clustering text documents into different category groups is an important step in indexing, retrieval, management and mining of abundant text data on the Web or in corporate information systems. However, current text clustering approaches tend to neglect several major aspects that greatly limit their practical applicability.

b) Problem Description

Text document clustering is a machine learning task taking place in a high dimensional space of word vectors, where each word, i.e. each entry of a vector, is seen as a potential attribute for a text. Empirical and mathematical analysis, however, has shown that in addition to computational inefficiencies clustering in high-dimensional spaces is very difficult because every data point tends to have the same distance from all other data points. To overcome from this drawback gather only the words that is very important for each document. But in some cases adding additional words (which may or may not contain in the document) for document vector gives better results when compared to vectors that do not contain additional words. Adding semantic information to each document using external sources like ontology's to document vector may increase the dimensionality, but it gives better clustering results. Efficient usages of this ontological information in preprocessing step are very crucial and plays important step in determining quality of the clusters. The quality of document clustering can be further improved by reducing the noise in the data in the pre-processing stage of data representation and also by applying some new clustering techniques.

c) Objective of the Project

The objective of current thesis work is to perform document clustering using WordNet (Ontology) derived information. Using this WordNet's information, a better representation in the form of document vectors for documents can be obtained. The document vectors form as input for performing document clustering. The other objective of the project is to study the relevance of bisecting k-means algorithm for document clustering compared to standard k-means algorithm.

The selected synsets of WordNet obtained during document preprocessing step contain of hierarchical conceptual information in a tree structure. The dissertation proposes to develop a GUI tool to represent the hierarchical WordNet concepts along with the facility to view the documents that belong to a particular concept. The tool assists the user in searching the documents based on generalized concepts to specific concepts. The clustering of documents formed at

each level of the concept tree can be viewed as a soft clustering as these clusters are overlapping.

2. Document Clustering

The manual extraction of patterns from data has occurred for centuries. Early methods of identifying patterns in data include Baye's theorem (1700s) and regression analysis (1800s). The increasing power of computer technology has increased data collection, storage and manipulations. As data sets have grown in size and complexity, direct hands-on data analysis has increasingly been augmented with indirect, automatic data processing. This has been aided by other discoveries in computer science, such as neural networks, cluster analysis, genetic algorithms (1950s), decision trees (1960s) and support vector machines (1990s). Data mining is the process of applying these methods to data with the intention of uncovering hidden patterns in large data sets.

Data mining a relatively young and interdisciplinary field of computer science is the process of discovering new patterns from large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics and database systems. The overall goal of the data mining process is to extract knowledge from a data set. Thus we can also say that Data mining is the process of extracting the implicit, previously unknown and potentially useful information from data.

The actual data mining task is analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection) and dependencies (association rule mining). These patterns can then be seen as a kind of summary of the input data, and used in further analysis. For example, the data mining step might identify multiple groups in the data, which can then be used to study the overall nature of the dataset. Thus instead of studying each record in the dataset, studying about the groups gives overall view of the dataset and saves lot of time.

Document clustering, which is one of the important areas of data mining is extensively used in the fields of text mining and information retrieval applications. Initially, document clustering was investigated for improving the precision in information retrieval systems and as an efficient way of finding the nearest neighbors of a document. More recently, clustering has been proposed for use in browsing a collection of documents or in organizing the results returned by a search engine in response to a user‘s query. Document clustering has also been used to automatically generate hierarchical clusters of documents.

2.1 Definition

2.1.1 Clustering

According to Brian Everitt Clustering means, ―Given a number of objects or individuals, each of which is described by a set of numerical measures, devise a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those from other classes. The number of classes and the characteristics of each class are to be determined

Uses of Clustering:

Clustering is the most common form of unsupervised learning and is an important in many fields of business and science. The following are some of the important areas in which clustering is useful.

- Search Optimization: Clustering helps in improving the quality and efficiency of search engines as the user query can be first compared to the clusters instead of comparing it directly to the documents and the search results can also be arranged easily.
- Finding Similar Documents: The interesting property here is that clustering is able to discover documents that are conceptually alike in contrast to search-based approaches that are only able to discover whether the documents share many of the same words.
- Organizing Large Document Collections: Document retrieval focuses on finding documents relevant to a particular query, but it fails to solve the problem of making sense of a large number of categorized documents. The challenge here is to organize these documents in a taxonomy identical to the one humans would create given enough time and use it as a browsing interface to the original collection of documents.
- Duplicate Content Detection: In many applications there is a need to find duplicates in a large number of documents. Clustering is employed for plagiarism detection, grouping of related news stories and to reorder search results rankings (to assure higher diversity among the topmost documents). Note that in such applications the description of clusters is rarely needed.
- Recommendation System: In this application a user is recommended articles based on the articles the user has already read. Clustering of the articles makes it possible in real time and improves the quality a lot.
- Clustering text documents into different category groups is an important step in indexing, retrieval, management and mining of abundant text data on the Web or in corporate information systems.

2.1.2 Introduction to Document Clustering

From the above applications like, finding similar documents, Organizing large Collections of Documents, search Optimization, indexing etc, we can conclude the importance of Document Clustering.

Document clustering is one of the important techniques of data mining which of unsupervised classification of documents into different groups (clusters), so those documents in each cluster share some common properties according to some defined similarity measure. So Documents in same cluster have high similarity but they are dissimilar to documents in other cluster. Some special requirements for good clustering algorithm:

- The document model should better preserve the relationship between words like synonyms in the documents since there are different words of same meaning.
- Associating a meaningful label to each final cluster is essential.
- The high dimensionality of text documents should be reduced.

For Document clustering, each document must be represented in some particular manner, so that it can be given as an input to the clustering algorithm. Bag of Words Representation is one such Representation.

2.1.3 Bag of Words Representation (BoW)

The BoW model allows a dictionary-based modeling, and each document looks like a "bag" (thus the order is not considered), which contains some words from the dictionary. Here are two simple text documents:

- Doc 1: John likes to watch movies. Mary likes too.
- Doc2: John also likes to watch football games.

Based on these two text documents, a dictionary is constructed as:
Dictionary= {1:"John", 2:"likes", 3:"to", 4:"watch", 5:"movies", 6:"also", 7:"football", 8:"games", 9:"Mary", 10:"too"}

This has 10 distinct words. And using the indexes of the dictionary, each document is represented by a 10-entry vector:

Doc 1: [1, 2, 1, 1, 1, 0, 0, 0, 1, 1]
Doc 2: [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

Where each entry of the vectors refers at particular index refers to the frequency of the corresponding word in that particular document. As we can see, this vector representation does not preserve the order of the words in the original sentences. Let us consider sample example. In the following example we have 9 documents and these documents are plotted on a two dimensional plane where X-axis represents frequency of term1 and Y-axis denotes frequency of term 2.

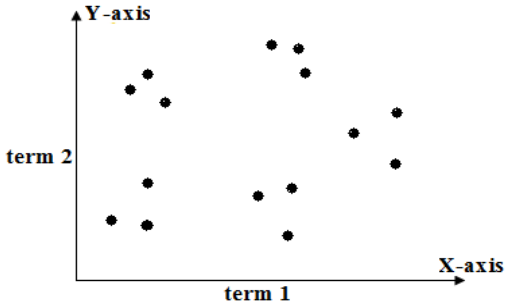

Figure 1: Documents plotted on two dimensional plane

From the Fig.1, 9 documents are plotted on the two-dimensional plane based on the frequencies of the two terms 'term 1' and 'term 2'. In Document clustering techniques we group documents which are near and generate clusters. After applying clustering algorithm on the above data for 5 clusters the output would be like the figure given in Fig.2.

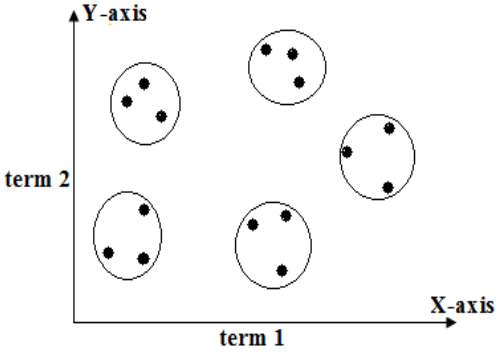

Figure 2.2: Clusters of the plotted documents

From the figures Fig.1 and Fig.2, it can be concluded that, given 15 documents are organized into 5 clusters. The documents within clusters have minimal distance among them but the distance between the documents of different clusters is high.

The major concern in information retrieval and text mining area is the question of finding the best method to explore and utilize the huge amount of text documents. Document clustering helps users to effectively navigate, summarize, and organize text documents. By organizing a large amount of documents into a number of meaningful clusters, document clustering can be used to browse a collection of documents or organize the results returned by a search engine in response to a user's query. Using clustering techniques to group documents can significantly improve the precision and recall in information retrieval systems and it is an efficient way to find the nearest neighbors of a document.

Text clustering typically is a clustering task working in a *high-dimensional space* where each word is seen as a potential attribute for a text. Empirical and mathematical analysis, however, has shown that in addition to computational inefficiencies— clustering in high-dimensional spaces is very difficult, because every data point tends to have the same distance from all other data points.

To overcome from this problem, reduce the dimension space by only considering features that are important for clustering and eliminate features that are infrequent. Since in high dimensional space every data point tends to have the same distance from all other data points, it is recommended to use cosine similarity as distance measure instead of Euclidean distance.

Before a set of documents can be presented to a machine learning system, each document must be transformed into a feature vector. Typically, each element of a feature vector represents a word from the corpus.

In order to reduce the dimensionality of the feature vector the removal of the stopwords from the feature vector is essential because these words do not add any information to the document. Therefore removing these stopwords from the feature vector reduces the dimensionality. The stopwords may include words like "the", "a", "an", "this", "what", "whose", "though" etc.

Stemming, stopword removal and pruning all aim to improve clustering quality by removing noise, i.e. meaningless data.

They all lead to a reduction in the number of dimensions in the term-space. Weighting is concerned with the estimation of the importance of individual terms.

The feature values may be binary, indicating presence or absence of the word in the document, or they may be integers or real numbers indicating some measure of frequency of the word's appearance in the text. This text representation, referred to as the *bag-of- words*, is used in most typical approaches to text classification. In these approaches, no linguistic processing (other than a stop list of most frequent words) is applied to the original text.

3. Ontology and Wordnet

In order to provide a more general form of cluster guidance, researchers have begun to investigate alternative clustering approaches that automatically incorporate background knowledge from external sources to guide the partitioning. By background knowledge we mean general information that can be exploited to improve clustering. One general approach to encoding background knowledge is the use of ontology.

Ontology is a hierarchy of concepts. In particular, there has been much work done on the use of semantic ontologies as an aid to clustering. In a semantic ontology, a concept is a word‘s sense. The relationships between senses are typically one of class–subclass. WordNet (http://wordnet.princeton.edu) and MeSH (http://www.nlm.nih.gov/mesh) are examples of existing semantic ontologies currently used in document clustering.

The term “ontology” has been used for a number of years by the artificial intelligence and knowledge representation community but is now becoming part of the standard terminology of a much wider community including information systems modeling. The term is borrowed from philosophy, where ontology means “a systematic account of existence”.

Ontology is “the specification of conceptualizations, used to help programs and humans share knowledge”. Ontology is a set of concepts - such as things, events, and relations that are specified in some way in order to create an agreed upon vocabulary for exchanging information. In information management and knowledge sharing arena, ontology can be defined as follows:

- Ontology is a vocabulary of concepts and relations rich enough to enable us to express knowledge and intention without semantic ambiguity.
- Ontology describes domain knowledge and provides an agreed-upon understanding of a domain.
- Ontology is collections of statements written in a language such as RDF that define the relations between concepts.

3.1 Terms and Definition

This section describes some of the commonly used terms along with their meaning with respect to ontology. Concept: An idea or thought that corresponds to some distinct entity or class of entities, or to its essential features, or determines the application of a term, and thus plays a part in the use of reason or language.

- Synset : Every word that is present in the ontology is called synset.
- Holonym: A concept of which this concept forms a part.
- Hypernym: Word with a broad meaning which more specific words fall under: a super ordinate.
- Hyponym: Word of more specific meaning.
- Meronym: A term that denotes part of something: a member of an information set.
- Ontology: The branch of metaphysics dealing with the nature of being.
- Semantic: Relating to meaning in language or logic.
- Synonym: A word or phrase that means exactly or nearly the same as another word.
- Whole: A term used to identify a concept that consists of multiple parts.

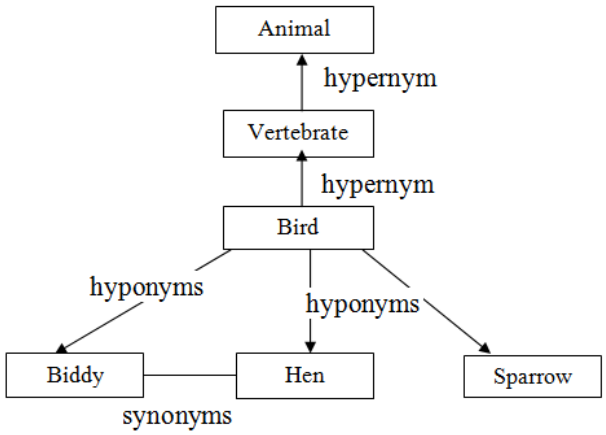

Figure 3: Example showing structure of Ontology

The figure *Fig.3* is a sample structure of synset representation in ontology. If we consider the synset “bird” in the above figure, it's more general synsets are “vertebrate” and one more level up is “animal”. Therefore these sysnets are called hypernyms to synset “bird”. For the synset “bird”, some more specific synsets are “biddy”, “hen”, “sparrow”; these 3 synsets are form of bird, hence called hyponyms to synset “bird”.

3.2 Benefits of Ontology

Ontology provides many benefits as listed below,

- To facilitate communications among people and organizations.
- To facilitate communications among system without semantic ambiguity.
- To provide foundations to build other ontology (reuse).
- To save time and effort in building similar knowledge systems (sharing).
- To make domain assumptions explicit.

3.3 Application Areas of Ontology

- Information Retrieval - As a tool for intelligent search through inference mechanism instead of keyword matching.

- Digital Libraries - Building dynamical catalogues from machine readable meta data, Automatic indexing and annotation of web pages or documents with meaning.
- Information Integration - Seamless integration of information from different websites and databases.
- Knowledge Engineering and Management -As a knowledge management tools for selective semantic access (meaning oriented access).
- Natural Language Processing - Better machine translation, Queries using natural language.

3.4 WordNet as Ontology

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonym sets. The database and software tools have been released under a BSD style license and can be downloaded and used freely. The database can also be browsed online. WordNet was created and is being maintained at the Cognitive Science Laboratory of Princeton University under the direction of psychology professor George A. Miller. Development began in 1985. WordNet's latest version is 3.1, as of June 2011. As of 2006, the database contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs; in compressed form, it is about 12 megabytes in size.

3.5 Structure of WordNet

The main relation among words in WordNet is synonymy, as between the words car and automobile. Synonyms words are the words that denote the same concept and are can be interchangeable in many contexts are grouped into unordered sets (synsets). Each of WordNet‘s 117,000 synsets is linked to other synsets by means of a small number of “conceptual relations”. Additionally, a synset contains a brief definition (“gloss”) and, in most cases, one or more short sentences illustrating the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets.

3.6 Relations in WordNet

The most frequently used relation among all the relations in synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation). It links more general synsets like {furniture} to increasingly specific ones like {bed}. Thus, WordNet states that the category furniture includes bed and conversely, concepts like bed make up the category furniture. All noun hierarchies ultimately go up the root node {entity}. Hyponymy relation is transitive, for example, if an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture. WordNet distinguishes among Types (common nouns) and Instances (specific persons, countries and geographic entities). Thus, armchair is a type of chair is an example of common noun whereas, Barrack Obama is an instance of a president is an example of specific noun. Instances are always leaf (terminal) nodes in their hierarchies.

WordNet also distinguishes between nouns, verbs, adjectives and adverbs because they follow different grammatical rules, it does not include prepositions, determiners etc.

4. WordNet Assisted Document Clustering

The process of document clustering in this thesis work is done using the semantics of document but not on the key word basis. To know the semantics of the document, an information source is necessary. Thus wordnet as ontology is used as source of information for the purpose of document clustering using semantics of the document.

4.1 WordNet Assisted Document Clustering using K-means algorithm

The input for this algorithm is set of documents that need to be clustered. For each document respected vector is generated. After generating vectors for all the documents, semantic based information is added to each vector. Once these preprocessing is finished, these vectors are given as input to k-means algorithm, specifying the number of clusters required. Once the algorithm finishes processing, it will give output clusters with documents that belong to that particular cluster. The efficiency of the algorithm can be analyzed using the Root Mean Square Error (RMSE).

4.2 Preprocessing of Documents

Clustering can be broken down into two stages. The first one is to preprocess the documents, i.e. transforming the documents into a suitable and useful data representation. The second stage is to analyze the prepared data and divide it into clusters, i.e. the clustering algorithm. Preprocessing the documents is probably at least as important as the choice of an algorithm, since an algorithm can only be as good as the data it works on.

4.2.1 Preprocessing Step:

Here we use the vector space model, in which a document is represented as a vector or ‘Bag of Words’, i.e., by the words it contains and their frequency, regardless of their order. A number of fairly standard techniques have been used to preprocess the data. In addition, a combination of standard and custom software tools have been used to add PoS tags and wordnet categories to the data set. The first preprocessing step is to PoS tag the corpus. The PoS tagger relies on the text structure and morphological differences to determine the appropriate part-of-speech. For this reason, if it is required, PoS tagging is the first step to be carried out. After this, stopword removal is performed, followed by stemming. This order is chosen to reduce the amount of words to be stemmed. The stemmed words are looked up in the wordnet and their corresponding synonyms and hypernyms are added to the bag-of-words. Once the document vectors are completed in this way, the frequency of the each word across the corpus can be counted and every word occurring is less often than the pre specified threshold is pruned. Finally, after the pruning step, the term weights are converted to *tf idf* as described below. Stemming, stopword removal and pruning all aim to improve clustering quality by removing noise, i.e. meaningless data. They all led to a reduction in a number of dimensions in the term-space.

Weighting is concerned with the estimation of the importance of individual terms. All of these have been used extensively and are considered the base line for comparison in this work. PoS tagging adds semantic information and wordnet is used to add synonyms and hypernyms. The rest of this section discusses preprocessing, clustering and evaluation in more detail.

1) Lexical analysis: lexical analysis of the text with the objective of removing digits,hyphens, punctuation marks from the documents and lower or upper case of letters is treated as same.
2) Stopword removal: stopwords are common words that do not add any semantic information to the text. For example: 'a', 'an', 'is', 'between' are all stopwords. Stopwords, i.e. words thought not to convey any meaning, are removed from the text. The approach taken in this work does not compile a static list of top words, as usually done. Instead PoS information is exploited and all tokens are not nouns, verbs are removed.
3) Collecting Nouns and Verbs: wordnet determines parts-of-speech for words. Collect all the nouns and verbs from the documents whose frequency is more than threshould.
4) Stemming: Words with the same meaning appear in various morphological forms. To capture their similarity they are normalized into a common root-form, the stem. The morphology function provided with WordNet is used for stemming, because it only yields stems that are contained in the WordNet dictionary.
   - A Stem: the portion of a word which is left after the removal of its affixes (i.e., prefixes or suffixes).
   - Example: connect is the stem for {connected, connecting connection, connections}

   Once the above steps are completed Corpus is generated.
5) *Corpus generation:* The corpus should contain global list of all nouns and verbs present in the document collection. A map is maintained for every synset that is present in the corpus, which stores the frequency of that particular synset in the corpus.
6) *Hypernym density representation:* In this phase, all nouns and verbs that are contained in the corpus are looked up in WordNet and a global list of all synonyms and hypernym synsets is assembled. Infrequently occurring synsets are discarded, and those that remain form the feature set. (A synset is defined as infrequent if its frequency of occurrence over the entire corpus is less than 0.05d, where d is the number of documents in the corpus.) For every word present in the corpus its frequency is calculated. Frequency is calculated by counting the number of times the word is present in the corpus. If 'd' is the number of documents present in the Document Collection, then Threshold (α) is defined as

   $$Threshold(\alpha)=0.05*d$$

The calculations of frequency and density are influenced by the value of a parameter h that controls the height of generalization (h). This parameter can be used to limit the number of steps upward through the hypernym hierarchy for each word.

- At height $h=0$ only the synsets that are contained in the corpus will be counted.
- At height $h>0$ the same synsets will be counted Which are at height h=0, as well as all the hypernym synsets that appear up to h steps above them in the hypernym hierarchy in the WordNet are also included.
- A value of h=h_max is defined as the level in which all hypernym synsets are counted, upto the top most root of the WordNet. At h=h_max, features corresponding to synsets higher up in the hypernym hierarchy represent supersets of the nouns represented by the less general features.

The best value of 'h' for a given text clustering task will depend on characteristics of the text such as use of terminology, similarity of topics, and breadth of topics. It will also depend on the characteristics of WordNet itself. In general, if the value for h is too small, the learner will be unable to generalize effectively. If the value for h is too large, the learner will suffer from overgeneralization because of the overlap between the features. Because of this, the experiment is repeated for different values of 'h' to understand the impact of generalization.

7) *Pruning:* Words appearing with low frequency(less than α) throughout the corpus are unlikely to appear in more than a handful of documents and would therefore, even if they contributed any discriminating power, be likely to cause too fine grained distinctions for us to be useful, i.e. clusters containing only one or two documents. Therefore all words that appear less often than a pre-specified threshold are pruned.
8) *Weighting:* Weights are assigned to give an indication of the importance of a word. The most trivial weight is the word-frequency. However, more sophisticated methods can provide better results. Throughout this work, *tf idf* (term frequency x inverse document frequency) is calculated. One problem with term frequency is that the lengths of the documents are not taken into account. The straight forward solution to this problem is to divide the term frequency by the total number of terms in the document, the document length. Effectively, this approach is equivalent to normalizing each document vector to length one and is called relative term frequency. However, for this research a more sophisticated measure is used: the product of term frequency and inverse document frequency *tf idf*.

   $$tf.idf(Wt,d)=\log(1+tft,d)*\log_{10}(N/dft)$$

   where term frequency *tf* is defined as the count of number of times the word is repeated in the vector.
   where *dft* is the number of documents in which term *t* appears and *N* the total number of documents.
   simply the multiplication of tf and idf. This means that larger weights are assigned to terms that appear relatively rarely throughout the corpus, but very frequently in individual documents. Thus these kinds of words have discriminating power and can discriminate from other documents.

   To overcome from the problem of over generalization we use idf value of the synset. In this process idf value is calculated for every synset according to the above mentioned formula. For synsets whose idf value is '1' indicates that the particular synset is present in every document. Therefore, such synset can be removed from

the corpus because such synsets does not give any weight to the clustering algorithm. Hence all the synsets from the hypernym density representation which has threshold less than (1<ß<2.5, since minimum value of tfidf is always 1 from the formula, where ß is user defined parameter) are removed.

Thus dimensionality can be further reduced by eliminating irrelevant synsets from the vectors. Now these vectors are given as input to the clustering algorithm where the number of dimensions is equal to the number of synsets that are present after pruning.

4.3 Document-term Matrix

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take. In this matrix, each row represents a document and each column represents the tfidf value of the corresponding synset for that particular document. This matrix is given as input to the Clustering Algorithm. Distance between different document vectors is calculated using Document-Term matrix. Table 4.1 depicts a document-term matrix for _n' documents and _m' synsets.

Table IV .1: Example of Term - Document matrix

F. no	Synset1	Synset2	…	…	Synsetm
Doc 1	23	42	25	34	45
Doc 2	13	56	54	21	88
…	…	…	…	…	…
Doc n	12	87	45	95	63(tfidf)

4.3.1 K-means clustering Algorithm

In data mining, *k-means clustering* is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. The term "*k*-means" was first used by James MacQueen in 1967, though the idea goes back to Hugo Steinhaus in 1957. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulse-code modulation, though it wasn't published until 1982. Simply speaking it is an algorithm to classify or to group your objects based on attributes/features into K number of group. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of K-mean clustering is to classify the data.

Algorithm
Input: Document Vectors DV
Number of Clusters 'k'
Output: 'k' Clusters Initially, the number of clusters must be known, or chosen, to be K say.

1. The initial step is the randomly choose a set of K instances as centers of the clusters.
2. Next, the algorithm considers each instance and assigns it to the cluster which is closest.
3. The cluster centroids are recalculated.
4. This process is iterated until there is not much change in the cluster centroids.

4.3.2 Distance Measure

Generally for high dimensional data instead of using Euclidean distance using cosine similarity gives better results, because Euclidian distance may intercept all the objects to be of equal distance for high dimensional data.

- Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them.
- The cosine of 0 is 1, and less than 1 for any other angle.
- The cosine of the angle between two vectors thus determines similarity between the vectors.

Given two vectors of attributes, *A* and *B*, the cosine similarity, *θ*, is represented using a dot product and magnitude as

$$similarly = \cos\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \, X \, B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} X \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

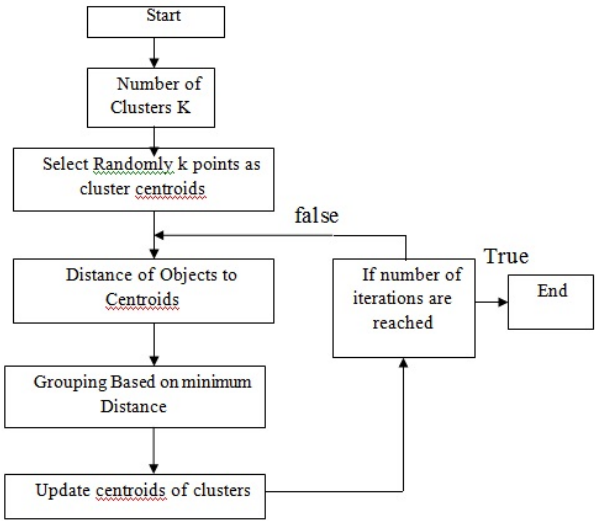

Figure IV.1: Flow chart of the k-means algorithm

Initially select randomly k points from the available points as initial seed points. Then assign each object to one of the cluster with nearest seed point from among the k points. Once each point is assigned to the nearest cluster, compute the centroid of the clusters. After computing the centroid of the cluster, repeat the above process again until required number of iterations is reached or the change in the cluster centoid is very small. In WADC_KA k-means algorithm is applied on document-term matrix generated using WordNet derived synset information. An overview of all the stages of Document Clustering can be explained from the below figures.

Stage 1: An overview of all the steps during preprocessing stage for generating Document Vectors

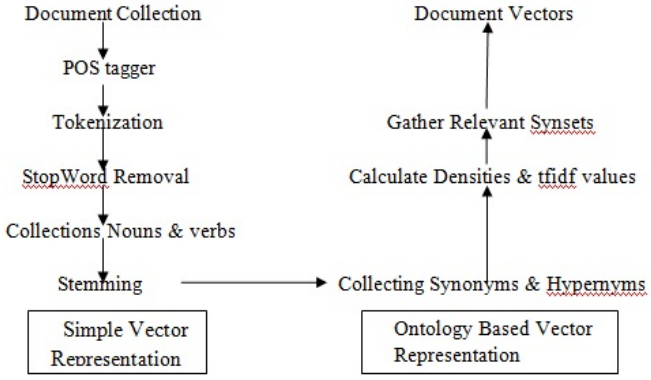

Figure IV.2: Steps during Preprocessing stage of Clustering

Figure Fig IV.2 explains about all the stages during the pre-processing step i.e., during the first step of clustering. This figure shows how the Document Collection is converted into Documents Vectors using WordNet Ontology as the source of information. Once this stage is completed, Document Vectors are given as input to the clustering Algorithms.

Stage 2: Performing Document Clustering using obtained Document Vectors

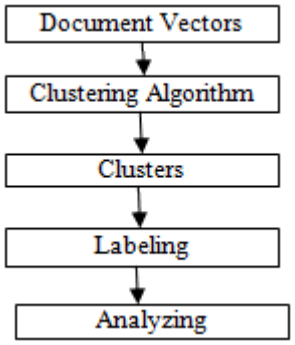

Figure IV.3: Clustering Document Vectors

The Figure 4.3 explains about the second stage during Document Clustering process. In this stage the clustering algorithm takes Document Vectors that are generated during the first stage, are considered as input to the K- means clustering Algorithm. Once clusters are generated they can be labeled and can be used for further analysis. Once the clusters are formed, labeling the cluster is important. Because once we label the cluster if will give a brief idea about the type of documents that are contained in that particular cluster.

Cluster Labeling: To name the cluster following steps are followed:

1. For each cluster, collect all the synsets along with its term frequency, for all the documents that are present in that particular cluster.
2. Arrange all the synsets according to the descending order of term frequency for the collected synsets.
3. Now label the cluster with the top most synsets. This gives an idea about the type of documents that are contained in the particular cluster.
4. Repeat the same procedure for all the clusters.

Problems of k-means Algorithm

- When number of features is more, k-means may not give good results.
- The user needs to specify *k*.
- K-means, the results vary quite a bit from one run to another.
- The algorithm is sensitive to outliers

1. Outliers are data points that are very far away from other data points.
2. Outliers could be errors in the data recording or some special data points with very different values.

But according to Michael Steinbach, it is suggested that bisecting k-means for Document Clustering is better than the regular k-means Clustering Algorithm. Hence Document Clustering is performed using Bisecting K-means to further improve results.

5. Improving Document Clustering Using Bisecting K- Means Algorithm

To improve the results obtained using K-means algorithm, other enhanced version of this algorithm called Bisecting K-means Algorithm can be implemented. Bisecting K- means Algorithm is also called as divisive hierarchical clustering algorithm. This is a top- down approach algorithm.

A. Introduction

The algorithm divides the dataset recursively into clusters. The *k*-means algorithm is used by setting k to two in order to divide the dataset into two subsets. Then the two subsets are divided again into two subsets by setting k to two. The recursion terminates when the dataset is divided into single data points or a stop criterion is reached. This process can be illustrated in the Fig V.*1*.

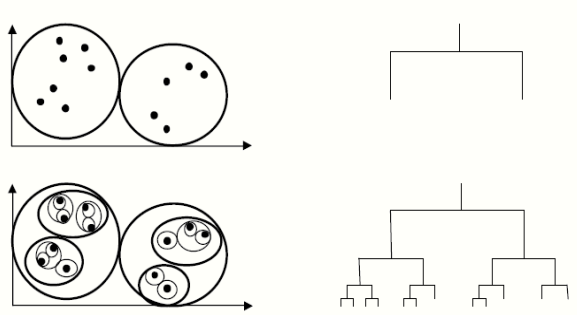

Figure V.1: Example of Bisecting K- means

Hierarchical *k*-means has O (n) run time. Such a run time is possible, because both the *k*-means algorithm and all operations concerning trees are possible in O (n).

While flat approaches create a flat partition of clusters, hierarchical clustering algorithms generate a dendrogram, i.e., a hierarchy in which each cluster is composed of two sub-clusters. Hierarchical clustering techniques can be classified into two sub-categories, divisive and agglomerative, on the basis of the approach used to create the hierarchy of clusters (either top-down or bottom-up).

Specifically, this algorithm also depends on the "bag of words" representation of textual documents. The objective of the evaluation was to be able to cluster large collections represented in a high dimensional and sparse space, which is a typical characteristics of information retrieval applications. However, the algorithm is general since it can cluster any type of items represented in a multidimensional vector space.

B. WordNet Assisted Document Clustering using Bisecting K-means algorithm

This algorithm resembles to a hierarchical clustering algorithm: in fact, hierarchical clustering algorithms have the advantage of not requiring a priori the number of clusters, since the clusters are bisected at each step. In these algorithms however, the problem is in defining a stopping rule, i.e., deciding if and which clusters have to be still bisected. To this aim, two main approaches are used: the first one applies the simple strategy of bisecting the greatest cluster and the second one is to split the cluster with greatest variance with respect to the centroid of the cluster.

Bisecting K-Means Algorithm:
Input: Document Vectors DV
Number of Clusters 'k'
Number of iterations of k-means ITER
Output: 'k' Clusters

1. Pick a cluster to split (split the largest).
2. Find 2 sub-clusters using the basic K-means algorithm.
3. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
4. Repeat steps 1, 2 and 3 until the desired number of clusters 'k' are reached.

In the above procedure ITER must be sufficiently large so that the change in the cluster centroid from its previous iteration is almost negligible.

Advantages of Bisecting K-means over K-means Algorithm:

1. Bisecting K-means tends to produce clusters of relatively uniform size. Because in every iteration, the cluster with maximum documents is selected for further bisection.
2. For bisecting K-means, there is not much change in the results from one run to another, whereas for regular K-means, the results vary quite a bit from one run to another.
3. Bisecting K-means gives better results for larger data sets.

6. Conclusion

Bisecting K-means tends to produce clusters of relatively uniform size whereas K-means produce clusters of non-uniform size. For bisecting K-means, there is not much change in the results from one run to another, whereas for regular K-means, the results vary quite a bit from one run to another. If the number of clusters is large, then bisecting K-means is more efficient than the regular K-means algorithm. Hence, Bisecting K-means gives better results for larger data sets. The RMSE values of Bisecting K-means have significant improvement over that of K-means based clustering.

References

[1] A.Hotho and S.Staab A.Maedche (2001), "Ontology-based Text Clustering", In proceedings of the IJCAI-2001 workshop Text Learning Beyond Supervision.
[2] Julian Sedding, "WordNet-based Text Document Clustering", Department of Computer Science, University of York Heslington, York YO10 5DD, United Kingdom.
[3] Michael Steinbach, George Karypis and Vipin Kumar(2001), "A Comparison of Document Clustering Techniques", Department of Computer Science and Engineering, University of Minnesota, Technical Report 00-034.
[4] Fellbaum, Christiane (2005), "WordNet and wordnets", In Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.
[5] WordNet http://wordnet.princeton.edu/Document-termmatrix, Wikipedia
[6] http://en.wikipedia.org/wiki/Document_clustering
[7] Document Clustering, Wikipedia http://en.wikipedia.org/wiki/Document_clustering
[8] DataMining, Wikipedia, http://en.wikipedia.org/wiki/ Data_ mining