An Efficient Semantic Model For Concept Based Clustering And Classification

SaiSindhu Bandaru[1]
Department of Information Technology
GVP College of Engineering
Visakhapatnam, India

Dr. K B Madhuri[2]
Department of Information Technology
GVP College of Engineering
Visakhapatnam, India

Abstract - Usually in text mining techniques the basic measures like term frequency of a term (word or phrase) is computed to compute the importance of the term in the document. But with statistical analysis, the original semantics of the term may not carry the exact meaning of the term. To overcome this problem, a new framework has been introduced which relies on concept based model and synonym based approach. The proposed model can efficiently find significant matching and related concepts between documents according to concept based and synonym based approaches. Large sets of experiments using the proposed model on different set in clustering and classification are conducted. Experimental results demonstrate the substantial enhancement of the clustering quality using sentence based, document based, corpus based and combined approach concept analysis. A new similarity measure has been proposed to find the similarity between a document and the existing clusters, which can be used in classification of the document with existing clusters.

Key words:
Label, Concept, PropBank, WordNet

I. INTRODUCTION

The recent techniques that are being used for text mining are concept-based which involve natural language processing as well as statistical analysis. Organization of existing documents and upcoming documents can be done by the mining functionalities clustering and classification. It is important and efficient if the classification functionality is embedded into the existing clustering functionality. Coming to natural language processing, to avoid the ambiguity of different senses of a single word and multiple representation for a single sense (depends on the author's vocabulary) NLP can be used. In the present work a new synonym based mining model has been proposed. It inherits all the benefits of existing concept based mining model. In addition to that it flavors the essence of synonym based matching. The present work models both clustering and classification at the same time the work shows that the same similarity measures can be used in synonym based approach also.

Text mining refers to the knowledge extraction from textual databases or documents. This text mining is different from mining the other types of databases because of its unstructured form and large number of dimensions. Each word in the document is a dimension. So the foremost things for text mining are, giving a structure to the data and reducing the dimensions.
Latent semantic analysis is a most popular method used in text mining. As the text data is unstructured data and higher dimensional data, the main things that has to be done in text mining are
1. Giving a structure to the unstructured data and
2. Reduce the dimensions as much as possible.

Giving structure to the data comes under natural language processing. Verb argument structure is one of the approaches for giving structure to a sentence. In this approach each word is given with a label (e.g. arg0, arg1 etc). There are different notations for these labels. This labeling can be done by semantic role parsing. That is, the label tells the semantic role of the word in that particular sentence.

Most of the text mining methodologies are based on vector space model. In this approach each text file is treated as a vector and the elements of vector are weight given to each word in that file.
Some methods used for text clustering include decision trees, conceptual clustering [3], clustering based on data summarization [4], statistical analysis neural nets, inductive logic programming, and rule-based systems among others.
The concepts can be identified by using natural language processing on the text document. That is by giving the structure to each sentence. This structure is called verb argument structure. See the example for verb argument structure of a sentence.

Example:
Sentence: He *hits* a ball.
Verb: hits
Arg0: he
Arg1: a ball

These labels are according to the prop bank notations [5]. A single word may have different senses. Using this semantic role, we can get the content in which the word is being used in that sentence.
Another important thing is a single sense can be represented by different words.

II. PROPOSED METHOD

The proposed mining model is an extension of the work in [1]. The proposed concept-based mining model consists of sentence-based concept analysis, document-based concept analysis, corpus-based concept-analysis, and concept-based similarity measure along with synonym based approach. A raw text document is the input to the proposed model. Each document has well-defined sentence boundaries. Each sentence in the document is labeled automatically based on the parts of speech. The number of generated labeled verb argument structures is entirely dependent on the amount of information in the sentence.

The sentence that has many labeled verb argument structures includes many verbs associated with their arguments. The labeled verb argument structures, the output of the role labeling task, are captured and analyzed by the concept-based mining model on sentence, document, and corpus levels. In this model, both the verb and the argument are considered as terms. One term can be an argument to more than one verb in the same sentence. This means that this term can have more than one semantic role in the same sentence. In such cases, this term plays important semantic roles that contribute to the meaning of the sentence. In the concept-based mining model, a labeled terms either word or phrase is considered as concept. The objective behind the concept-based analysis task is to achieve an accurate analysis of concepts on the sentence, document, and corpus levels rather than a single-term analysis on the document only.

SEMANTIC AND CONCEPT-BASED MINING MODEL

The proposed mining model can be depicted as shown in figure 1.

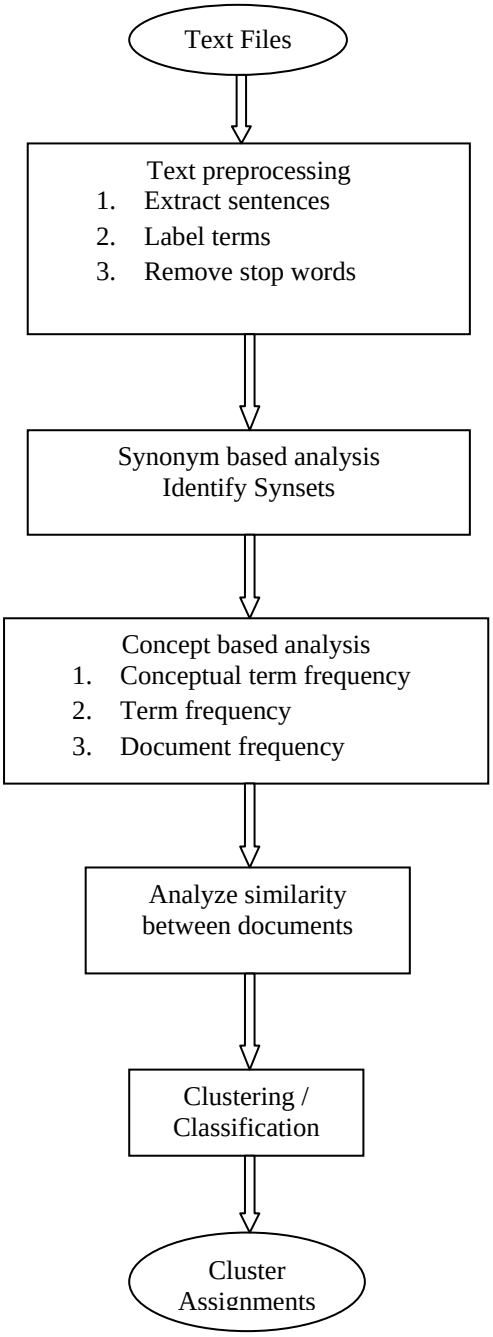

Figure 1. Mining Model

.

The sentence that has many labeled verb argument structures includes many verbs associated with their arguments. The labeled verb argument structures, the output of the role labeling task, are captured and analyzed by the concept-based mining model on sentence, document, and corpus levels. In this model, both the verb and the argument are considered as terms. One term can be an argument to more than one verb in the same sentence. This means that this term can have more than one semantic role in the same sentence. In such cases, this term plays important semantic roles that contribute to the meaning of the sentence. In the concept-based mining model, a labeled terms either word or phrase is considered as concept. The objective behind the concept-based analysis task is to achieve an accurate analysis of concepts on the sentence, document, and corpus levels rather than a single-term analysis on the document only.

Step wise process:

- Preprocessing of text
- Identify the concepts
- Calculating conceptual term frequency at
    - Sentence level
    - Document level
    - Corpus level
- Find Synsets for each concept
- Identify significant concepts based on frequency
- Cluster the documents

Preprocessing of text :

In this module each document is read from the corpus. In each document, the sentences are separated. As the raw text data is unstructured data, we have to give a proper structure to each sentence. So each sentence is given a verb argument structure.
To get the verb argument structure, each word in the sentence is tagged with the parts of speech of that word in that sentence. Using these parts of speech for each term the verbs are identified.
And then for each verb arguments are identified. These arguments are labeled as ARG0, ARG1, ARG2 etc. basing on the number of verbs for which the term is argument. For example if a term is argument for a single verb then the argument is ARG0. If it is argument for two verbs then the label is ARG1.
Another important technique in text mining is reducing the dimensionality of the text. That is we have to remove some unnecessary words. This can be done using standard stop lists. Each word is checked against the standard stop word list. If it is a stop word, then it is treated as insignificant word and it is removed from the process.

Identify the concepts:
After completion of first step, we are remained with the labeled terms which are significant for matching that is to find the similarity. So each labeled term is treated as a concept.

Calculating conceptual term frequency:
To analyze each concept at the sentence level, a new concept-based frequency measure, called the conceptual term frequency *ctf* is used [1]. The ctf calculations of concept c in sentence s and document d are as follows:

At sentence level(conceptual term frequency ctf):
The *ctf* is the number of occurrences of concept c in verb argument structures of sentence s. The concept c, which frequently appears in different verb argument structures of the same sentence s, has the principal role of contributing to the meaning of s. In this case, the *ctf* is a local measure on the sentence level.
A concept c can have many ctf values in different sentences in the same document d. Thus, the ctf value of concept c in document d is calculated by

$$ctf = \frac{\sum_{n=1}^{sn} ctf_n}{n} \text{ - - - - - - - - - - - - (1)}$$

where sn is the total number of sentences that contain concept c in document d. Taking the average of the ctf values of concept c in its sentences of document d measures the overall importance of concept c to the meaning of its sentences in document d. A concept, which has ctf values in most of the sentences in a document, has a major contribution to the meaning of its sentences that leads to discover the topic of the document. Thus, calculating the average of the ctf values measures the overall importance of each concept to the semantics of a document through the sentences.

At document level(term frequency tf)
To analyze each concept at the document level, the concept- based term frequency tf, the number of occurrences of a concept (word or phrase) c in the original document, is calculated. The tf is a local measure on the document level.

At corpus level(document frequency df):
To extract concepts that can discriminate between documents, the concept-based document frequency df, the number of documents containing concept c, is calculated. The df is a global measure on the corpus level. This

measure is used to reward the concepts that only appear in a small number of documents as these concepts can discriminate their documents among others.

Find Synsets for each concept:

Depending on the author's vocabulary to have the same semantics different words may be used. For example "intelligent" and "brilliant". Both are of same meaning but words are different. To identify theses type of words we have to find the synonyms for each concept. We have an efficient data base called WordNet which gives synonyms for words. The set of synonyms for a word is called synset. So by using word net database we can get synset for each concept. While finding the matching between documents words are being compared. When there is no exact word matching then the corresponding synsets will be checked for matching. So the original semantic are preserved.

Identify significant concepts based on frequency:
Based on the frequency at three levels , weightage will be given to each concept. The more significant concept will have more weight.
The weights can be calculated as follows.

$$weight_i = (tf\ weight_i + ctf\ weight_i) * log\left(\frac{N}{df_i}\right) \quad --(3)$$

$$tf\ weight_i = \frac{tf_{ij}}{\sqrt{\sum_{j=1}^{cn}(tf_{ij})^2}} \quad ----(4)$$

$$ctf\ weight_i = \frac{ctf_{ij}}{\sqrt{\sum_{j=1}^{cn}(ctf_{ij})^2}} \quad -----(5)$$

1. the total number of documents, N, in the corpus.
2. the ctf_{ij} of each concept c_i in s for each document d_j.
3. the tf_{ij} of each concept c_i in each document d_j.
4. the df_i of each concept c_i.
5. *cn* is the total number of the concepts which has a term frequency value in document d.

Cluster documents:

To cluster the documents we need a clustering algorithm. K Nearest Neighbors algorithm is a most popular algorithm that can be used for any type of clustering process. In text clustering each text file is considered as a data point and similarity between two documents is treated as distance between two data points.
So for each pair of documents in the corpus, similarity is calculated. A similarity matrix is built to represent the similarity between each pair of documents in the corpus.
The similarity between two documents d_1 and d_2 is calculated using the formula.

$$sim_v(d_1, d_2) = \sum_{i=1}^{m}\left(\frac{l_{i_1}}{Lv_{i_1}}, \frac{l_{i_2}}{Lv_{i_2}}\right) \times weight_{i_1} \times weight_{i_2} \quad (6)$$

Where d_1 and d_2 two documents,
the number of matching concepts, m, in the verb argument structures in each document d,
the total number of sentences, sn, that contain matching concept c_i in each document d,
the total number of the labeled verb argument structures, v, in each sentence s,
the length, *l*, of each concept in the verb argument structure in each document d,
the length, *Lv*, of each verb argument structure which contains a matched concept, and
$weight_i$ can be calculated using equation (2).

The algorithm for finding the similarity is given below. This algorithm takes two documents as input. And it calculates the similarity between these documents using the formula given in equation (5).

III. CLASSIFICATION

Once clusters are formed with existing corpus, classification of upcoming files can be done. That is we can assign each upcoming file into any one of the existing cluster if there is match the properties of cluster and file. Otherwise a new cluster can be formed with this new file.

So to classify a new file, first this file will be preprocessed as said in previous section.
Once concepts are identified, then matching process will be performed with each document in each cluster. So the similarity between each file in each cluster will be calculated. The new file will be assigned into the cluster for which cluster this new file has similarity with more number of files.
If the new file is not similar with any of the existing clusters then a new cluster will be formed with only one member that is the new file.

Algorithm2: for classification:
Input: set of files D, set of cluster ids C, clusters
Output: cluster id assignments to each file
Procedure:
Begin
Let C be array of cluster ids
Let M be an array of size equals to no of clusters
for each file f_d in D
for each cluster C_i
M[i]= get_similarity(f_d, C_i)
end for
cid=find_max(M)
output cid as cluster id for file f_d
update clusters // assign f_d to cluster
with cid as cluter id if cid is new
form new cluster
end for
end
find_max(M)
{
Let cid=0;
for i 1 to size(M)
if m[i]>m[cid]
cid=i;
end for
if(cid=0)
cid=size(M)+1; // new cluster id is formed
return cid;
}

In the process of classification, the cluster id for which the given file is more similar has to be identified. This can be done using the above mentioned algorithm. A new similarity measure has been introduced to get the similarity between a single file and a cluster. this similarity measure is a function of similarity between the given document and each document in the cluster and total number of files in the cluster.

$$sim(doc, C_i) = \sum_{i=0}^{n} weight_i \quad (6)$$

$$weight_i = \frac{sim(doc, d_i)}{\sum_{j=0}^{n} sim(doc, d_j)} \quad (7)$$

Where

1. *doc* is the document that has to be classified.
2. C_i is the ith cluster.
3. n is the total numbers of files in cluster C_i.
4. d_i is the ith file in the cluster C_i.
5. $sim(doc, d_i)$ is calculated using equation (2).

get_similarity () returns the similarity between a document and the cluster. After finding similarity with all clusters, then the cluster id for which similarity is more, is assigned as cluster for the new document.

IV. ADVANTAGES OF PROPOSED METHOD

In this new frame work proposed, the term matching is done very efficiently.

Maintaining synonym vector for each concept is costly in term of space complexity. So here only if the exact word match is not found then only the synonym list will be identified. And then the matching process will be continued with the synonym list.

Another advantage is it will perform both classification and clustering also. While performing the classification if one file is not similar to the files in existing clusters then a new cluster will be formed with only one file.

This clustering process is adopting both agglomerative and hierarchical features.

While finding the similarity there is no need of considering any threshold given by user. The threshold will be taken automatically depending on similarity measures calculated at that instance.

V . EXPERIMENTAL RESULTS

To test the effectiveness of the proposed mining model the standard data set 20 News Group Data Set is taken. Total 100 documents of 5 different groups have been taken for testing.
The measure Precision, Recall and F-score are calculated after clusters are formed

Measure	Value(average for all clusters) (Existing Method)	Value (average for all clusters) (Proposed Method)
Precesion	0.54	0.625
Recall	0.58	0.8
F-score	0.58	0.66

$$Precesion(i,j) = \frac{M_{ij}}{M_j}$$ ----------(8)

$$Recall(i,j) = \frac{M_{ij}}{M_i}$$ ----------(9)

$$F - score = \frac{2PR}{P+R}$$ ----------(10)

where M_{ij} is the number of members of class i in cluster j, M_j is the number of members of cluster j, and M_i is the number of members of class i.

CONCLUSION

This frame work is utilizing the features of both natural language processing and statistical techniques of data mining approaches. It is using the benefits of three level similarity measures that are sentence level, document

level and corpus level. The results have shown that incorporation of semantic approach is resulting 20% improvement in f-measure values for the process of clustering. As an extension, this mining model is useful to perform the classification operation on files also.

This work can be extended in a way that it can be combined with data summarization methodologies to improve time considerations and to speed up the processing of documents.

REFERENCES

[1] Shady Shehata, , Fakhri Karray, and Mohamed S. Kamel, "An Efficient Concept-Based Mining Model for Enhancing Text Clustering," IEEE Transactions on Knowledge and Data Engineering , vol. 22, no. 10, October 2010.

[2] L.Talavera and J. Bejar, "Generality-Based Conceptual Clustering with Probabilistic Concepts," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 196-206, Feb. 2001.

[3] H.Jin, M.-L. Wong, and K.S. Leung, "Scalable Model-Based Clustering for Large Databases Based on Data Summarization," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 1710-1719, Nov. 2005

[4] P.Mitra, C. Murthy, and S.K. Pal, "Unsupervised Feature Selection Using Feature Similarity," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 301-312, Mar. 2002.

[5] Olga Babko-Malaya "Propbank Annotation Guidelines" pp. 2-19 September 2005.

[6] Kowalski, Gerald, Mark T Maybury: Information Retreival Systems: Theory and Implementation, Kluwer Academic press 1997.

[7] Data mining – Concepts and Techniques – Jlaweihan & Micheline Kamber Harcourt India,

[8] pp.614-627,2008

[9] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles," Computational Linguistics, vol. 28, no. 3, pp. 245-288, 2002

[10] S.Y. Lu and K.S. Fu, "A Sentence-to-Sentence Clustering Procedure for Pattern Analysis", IEEE Trans. Systems, Man, and Cybernetics, vol. 8, no. 5, pp. 381-389, May 1978.