

Application of Concept-Based Mining Model in Text Clustering

Modu Sowjanya

*M.Tech Student
Dept of cse,AIET*

K.Ravindra

*Asst. Professor,
Dept of CSE, AIET*

Y.Ramesh Kumar

*Asst. Professor,
Dept of CSE, AIET*

Abstract:Most of the common techniques of text mining are based on the statistical analysis of the term frequency. The statistical analysis of the term frequency captures the importance of the term within the document only. An alternate approach would be to enhance the mining model to include the contribution of the term to the semantics of the text so that the terms that capture the concepts of the document and thereby the similarity between the documents may be found. The contribution of each term to the semantics at the sentence, document and corpus levels is determined using sentence-based concept-analysis, document-based concept-analysis and corpus-based concept-analysis respectively. A concept-based similarity measure is used to determine the similarity between the documents. This paper introduces the Concept-based similarity measure into the Temporal -Semantic Clustering model for event detection in newspaper articles. The document similarity function is defined in terms of two similarity measures. Initially a context based similarity measure that uses the vector of weighted terms is used to determine the similarity between the documents. Later another partial similarity measure that uses the vector of weighted time entities along with the previously determined concept-based measure to determine the combined similarity measure. Hierarchical approach is used to cluster documents based on this similarity measure.

Keywords: Temporal-Semantic Clustering, Concept-based clustering, Text Mining

1. INTRODUCTION

Data Mining refers to extracting or mining knowledge from large amounts of data. Data Mining is also treated as an essential step in Knowledge Discovery in Databases or KDD.

- Data Cleaning
- Data Integration
- Data Selection
- Data Transformation
- Data Mining
- Pattern Evaluation
- Knowledge Presentation

Text mining is the process of extracting important information and knowledge from unstructured text. The main difference between the text mining and the data mining is that data mining tools are designed to deal with structured data from databases or XML-based tables. However, text mining deals with unstructured or semi-structured data such as text documents, HTML tables, and emails. Thus, text mining is a much generalized solution for text, where large volumes of different types of information should be managed and merged.

The purpose of Text Mining is to process unstructured (textual) information, extract meaningful numeric indices

from the text, and, thus, make the information contained in the text accessible to the various data mining (statistical and machine learning) algorithms. Information can be extracted to derive summaries for the words contained in the documents or to compute summaries for the documents based on the words contained in them. Hence Text mining makes it possible to analyze words, clusters of words used in documents. Further analyzing documents and determining similarities between them or how they are related to other variables of interest in the data mining project can also be done using text mining. In the most general terms, text mining will "turn text into numbers" (meaningful indices), which can then be incorporated in other analyses such as predictive data mining projects, the application of unsupervised learning methods (clustering).

In data mining, usually there is a fixed model for data that is used by most mining algorithms. This data model varies depending on the nature of the data. One of the most broadly used models in text mining systems is the Vector Space Model (VSM). Documents and terms are represented in a high dimensional space. Each dimension corresponds to a word in the documents collection. The documents vectors that are closed to the term vector represent the most relevant documents to the term which means that these documents contain words similar to the words in the term. When using VSM, the feature space constitutes a metric space where documents represented as points in a multidimensional space. There are two measures that are usually used which are the cosine measure, and the Jaccard measure.

Document clustering aims to automatically divide documents into groups based on similarities of their contents. Each group (or cluster) consists of documents that are similar between themselves (have high intra-cluster similarity) and dissimilar to documents of other groups (have low inter-cluster similarity). Clustering documents can be considered as an unsupervised task that attempts to classify documents by discovering underlying patterns, i.e., the learning process is unsupervised, which means that no need to define the correct output (i.e., the actual cluster into which the input should be mapped to) for an input.

Approaches to Text Mining

Text mining can be summarized as a process of "numeric zing" text. At the simplest level, all words found in the input documents will be indexed and counted in order to compute a table of documents and words, i.e., a matrix of frequencies that enumerates the number of times that each word occurs in each document. This basic process can be further refined to exclude certain common words such as "the" and "a" (stop word lists) and to combine different

grammatical forms of the same words such as "traveling," "traveled," "travel," etc. (stemming). However, once a table of (unique) words (terms) by documents has been derived, all standard statistical and data mining techniques can be applied to derive dimensions or clusters of words or documents, or to identify "important" words or terms that best predict another outcome variable of interest.

Using well-tested methods and understanding the results of text mining: Once a data matrix has been computed from the input documents and words found in those documents, various well-known analytic techniques can be used for further processing those data including methods for clustering.

"Black-box" approaches to text mining and extraction of concepts: There are text mining applications which offer "black-box" methods to extract "deep meaning" from documents with little human effort (to first read and understand those documents). These text mining applications rely on proprietary algorithms for presumably extracting "concepts" from text, and may even claim to be able to summarize large numbers of text documents automatically, retaining the core and most important meaning of those documents. While there are numerous algorithmic approaches to extracting "meaning from documents," this type of technology is very much still in its infancy, and the aspiration to provide meaningful automated summaries of large numbers of documents may forever remain elusive. If it is not clear to the user how those algorithms work, it cannot possibly be clear how to interpret the results of those algorithms. And the methods used in those programs are not open to scrutiny by the academic community and peer review and, hence, we simply don't know how well they might perform in different domains. As a final thought on this subject, you may consider this concrete example: Try the various automated translation services available via the Web that can translate entire paragraphs of text from one language into another. Then translate some text, even simple text, from your native language to some other language and back, and review the results. Almost every time, the attempt to translate even short sentences to other languages and back while retaining the original meaning of the sentence produces humorous rather than accurate results. This illustrates the difficulty of automatically interpreting the meaning of text.

Text mining as document search: There is another type of application that is often described and referred to as "text mining" - the automatic search of large numbers of documents based on key words or key phrases. This is the domain of, for example, the popular internet search engines that have been developed over the last decade to provide efficient access to Web pages with certain content. While this is obviously an important type of application with many uses in any organization that needs to search very large document repositories based on varying criteria, it is very different from what has been described here.

Text Clustering: Common Text Clustering systems take into consideration statistical analysis of a term, i.e., the frequency of a term (word or phrase) within a document to explore the importance of a term within the document.

However, two terms can have the same frequency in their documents, but one term may contribute more to the meaning of its sentence rather than the other term. Thus the semantic structure of the sentences in the document is not taken into consideration and the quality of clustering suffers.

Single pass clustering method expects a similarity matrix as its input and outputs clusters. The clustering method takes each object sequentially and assigns it to the closest previously created cluster, or creates a new cluster with that object as its first member. A new cluster is created when the similarity to the closest cluster is less than a specified threshold. This threshold is the only externally imposed parameter. Commonly, the similarity between an object and a cluster is determined by computing the average similarity of the object to all objects in that cluster.

The rapid rise in the number of news articles has made the task of discovering, tracking and organizing the topics associated with them difficult. For each incoming document detecting an event consists of determining whether it reports a new event or an older event that has occurred before. All the currently used Topic Detection and Tracking (TDT) systems use a document clustering algorithm along with the date of publication of the news articles. Identifying the topics available in the documents and grouping the events based on the temporal information available in them is the main aim of the TDT Systems. In this paper the hierarchical clustering algorithm presented in [1] is

used. Documents that have high temporal-semantic similarity are grouped together based on the modified similarity function. This similarity function is based on the temporal-semantic similarity function of [1] which in turn uses the concept-based similarity function presented in [2] instead of using the cosine measure. In this paper the hierarchical clustering algorithm presented in [1] is used. At the initial level documents with high temporal-semantic similarity are grouped together using the modified document similarity function to detect the occurrence of an event. Grouping of documents at later levels will result in more complex events and topics being identified. This results in the documents being clustered basing on the content as well as temporal occurrences available in the documents.

2. METHODOLOGY

2.1. Concept-based Mining Model

Concepts convey local context information, which is essential in determining an accurate similarity between documents. A concept-based similarity measure, based on matching concepts at the sentence, document, corpus and combined approach rather than on individual terms (words) only, is devised. The concept-based similarity measure relies on three critical aspects. First, the analyzed labeled terms are the concepts that capture the semantic structure of each sentence. Secondly, the frequency of a concept is used to measure the contribution of the concept to the meaning of the sentence, as well as to the main topics of the document. Lastly, the number of documents that contains the analyzed concepts is used to discriminate among documents in calculating the similarity. These aspects are

measured by the proposed concept-based similarity measure which measures the importance of each concept at the sentence-level by the *ctf* measure, document-level by the *tf* measure, and corpus-level by the *df* measure. The concept-based measure exploits the information extracted from the concept-based analysis algorithm to better judge the similarity between the documents.

The Concept-based Mining Model system is a text mining application that uses the concept based similarity measure to determine the similarity measure between the documents. A raw text document is input to the proposed system by the user. Each document has well defined sentence boundaries. Each sentence in the document is labeled automatically based on the PropBank notations. After running the semantic role labeler, each sentence in the document might have one or more labeled verb argument structures. The number of generated 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 the sentence, document and corpus levels.

In this model both the verb and argument are considered as *terms* and labeled terms are considered *concepts*. 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 and hence it contributes more to the meaning of the sentence.

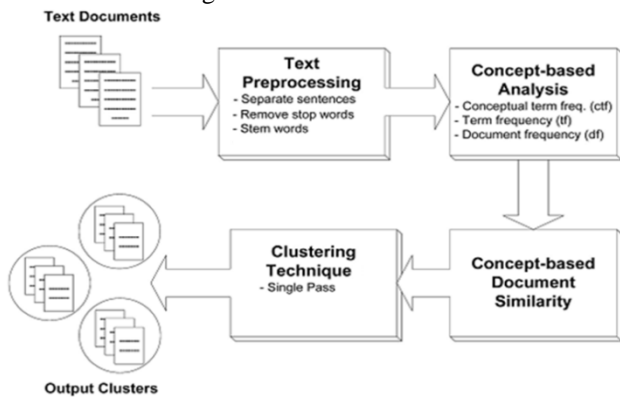


Fig1. Concept- based mining model.

2.1.1.Sentence – based concept analysis

To analyze each concept at the sentence level, the concept-based frequency measure called the conceptual term frequency (*ctf*) is proposed. The *ctf* calculations of concept *c* in sentence *s* and document *d* are as follows:

Calculating *ctf* of Concept *c* in Sentence *s* : 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* will have a larger contribution to make to the meaning of *s*. Thus, the *ctf* measure is a local measure on the sentence level.

Calculating *ctf* of Concept *c* in Document *d* : A concept 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:

$$ctf = \frac{\sum_{n=1}^{sn} ctf_n}{sn}$$

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 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 the document through its sentences.

2.1.2. Document-Based Concept Analysis

To analyze the 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.

2.1.3. Corpus-Based Concept Analysis

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 appear only in a small number of documents as these concepts can discriminate their documents among others.

The process of calculating *ctf*, *tf* and *df* measures in a corpus is attained by the proposed algorithm which is called Concept-based Analysis algorithm.

2.1.4. Concept-Based Analysis algorithm

1. d_{doci} is a new Document
2. *L* is an empty List (*L* is a matched concept list)
3. S_{doci} is a new sentence in d_{doci}
4. Build concepts list C_{doci} from S_{doci}
5. for each concept $c_i \in C_i$ do
6. compute ctf_i of c_i in d_{doci}
7. compute tf_i of c_i in d_{doci}
8. compute df_i of c_i in d_{doci}
9. d_k is seen document, where $k = \{0, 1, \dots, d_{doci} - 1\}$
10. s_k is a sentence in d_k
11. Build concepts list C_k from s_k
12. for each concept list $c_j \in C_k$ do
13. if ($c_i = c_j$) then
14. update df_i of c_i
15. compute $ctf_{weight} = avg(ctf_i, ctf_j)$
16. add new concept matches to *L*
17. end if
18. end for
19. end for
20. Output the matched concepts list *L*.

The concept-based analysis algorithm describes the process of calculating *ctf*, *tf* and *df* values of the matched concepts in the documents. The procedure begins with processing a new document which has well defined sentence boundaries. Each sentence is semantically labeled. The lengths of the matched concepts and their verb argument structures are stored for the concept-based similarity calculations.

Each concept in the verb argument structures which represents the semantic structure of the sentence is processed sequentially. Each concept in the current document is matched with the other concepts in the previously processed documents. To match the concepts in the previous documents a concept list L which holds an entry for each of the previous documents that shares a concept with the previous documents is maintained. After the document is processed, L contains all the matching concepts between the current document and any previous document that contains atleast one matching concept with the new document.

2.1.5. Concept-based Similarity measure

The concept-based measure exploits the information extracted from the concept-based algorithm to better judge the similarity between the documents. The similarity measure is a function of the following factors:

- A. No. of matching concepts m in the verb argument structures in each document d
- B. Total no. of sentences sn that contain matching concept ci in each document d
- C. Total no. of labeled verb argument structures v, in each sentence s
- D. The ctfi of each concept ci in s for each document d, where i=1,2,3,...,m
- E. The tfi of each concept ci in each document d, where i=1,2,3,...,m
- F. The dfi of each concept ci in each document d, where i=1,2,3,...,m
- G. The length l of each concept in the verb argument structure in each document d
- H. The length Lv of each verb argument structure which contains a matched document, and The total no. of documents, N, in the corpus

The concept based similarity between two documents d1 and d2 is calculated by:

$$sim_c(d_1, d_2) = \sum_{i=1}^m \max\left(\frac{tf_{i1}}{l_{i1}}, \frac{tf_{i2}}{l_{i2}}\right) \times weight_{i1} \times weight_{i2}$$

The concept based weight of concept i in document d is calculated by:

$$weight_i = (tfweight_i + cfwweight_i) \times \log\left(\frac{N}{df_i}\right)$$

The tfweighti, cfwweighti values represent the weight of concept i in document d at document level and sentence level respectively.

The log(N/dfi) value rewards the weight of the concept i on the corpus level when concept i occurs in small no. of documents.

The tfij and ctfij values are normalized by the length of the document vector of term frequency and conceptual term frequency of document d as:

$$tfweight_i = \frac{tf_{ij}}{\sqrt{\sum_{j=1}^m (tf_{ij})^2}}$$

$$cfweight_i = \frac{ctf_{ij}}{\sqrt{\sum_{j=1}^m (ctf_{ij})^2}}$$

3. DATA ANALYSIS

We were considered some text data and entered some keywords and delete those keywords from text data. The data as shown in the below figure2

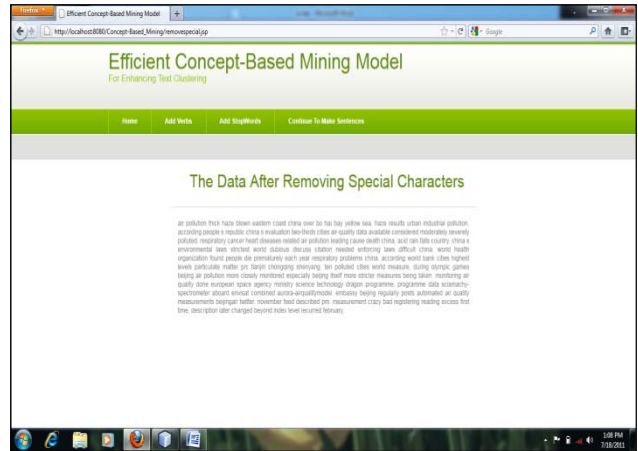


Figure2

Sentences formed by the data is as shown in figure3

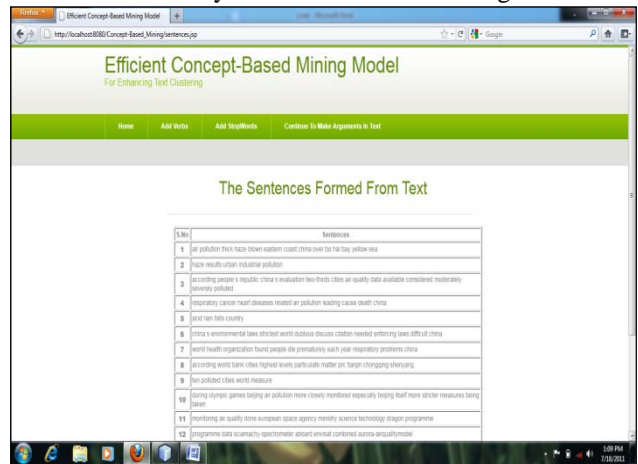


Figure3

Arguments of the text as shown in the figure 4.

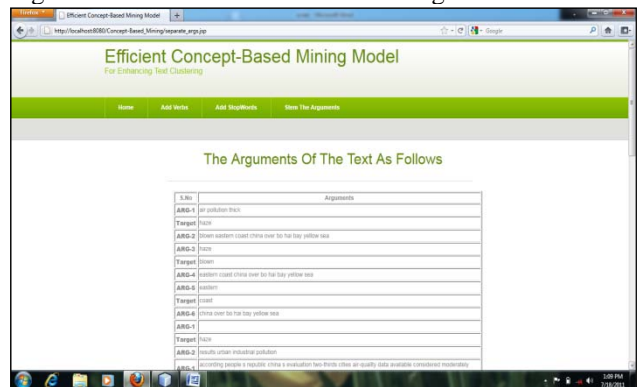


Figure4

The conceptual frequency, term frequency and document frequency as shown in the figures 5,6,7 respectively

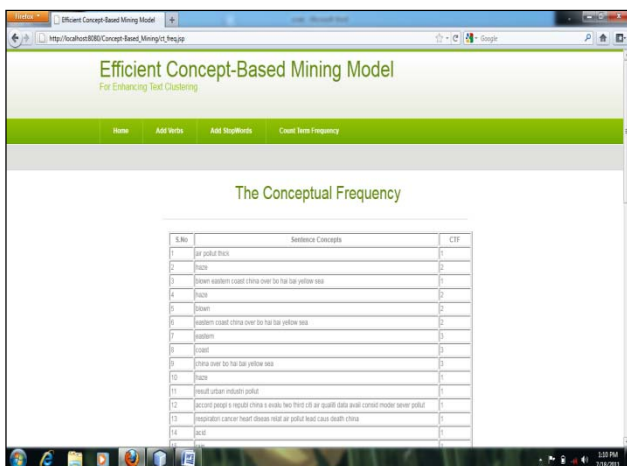


Figure5

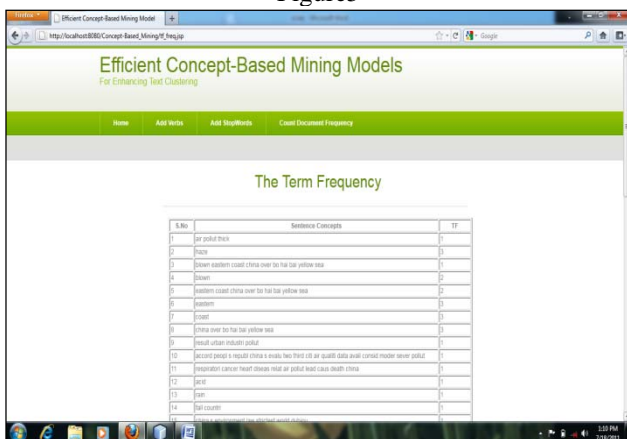


Figure6

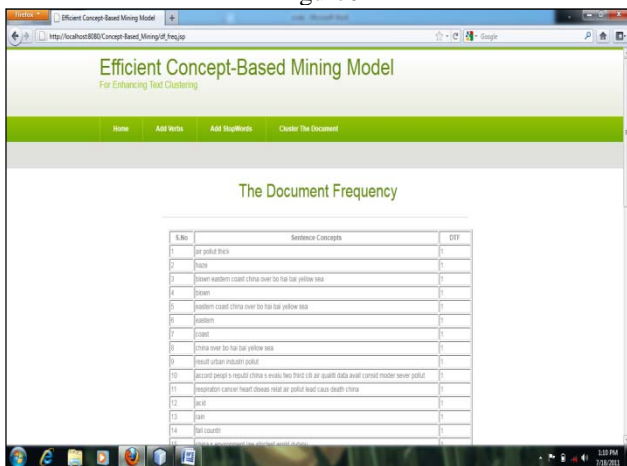


Figure7

CONCLUSION

In this project we tried to apply the concept-based approach to text clustering. The proposed system exploited fully the semantic structure of the sentences in the documents in order to achieve good quality of clustering. To the input document Text pre-processing was initially done where the sentences were separated and labeled with verb argument structures. Further stop words were removed and stemming was done. This was followed by components that performed sentence-based, document-based, corpus-based and concept-based analysis where the conceptual term

frequency measure (ctf), concept-based term frequency measure (tf), document term frequency measure (df) and the concept-based similarity measure were determined respectively. Finally clustering of document was done. If the similarity measure resulted in a value that was less than the threshold it was placed into the same cluster, otherwise it placed into a separate cluster.

The implementation of the concept –based similarity function can be used in applications where document similarity is used to cluster the documents for example in applications that cluster newspaper articles for topic detection and tracking. The concept based similarity function can be used for web document clustering.

REFERENCES

- [1] An Efficient Concept-Based Mining Model for enhancing Text Clustering, Shady Shehata, FakhriKarray, Mohamed S.Kamel, IEEE Transactions on knowledge and Data Engineering, vol 22, no 10, October 2010.
- [2] K.J. Cios, W. Pedrycz, and R.W. Swinarski, Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998.
- [3] B. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures and Algorithms. Prentice Hall, 1992.
- [4] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical Report 941, Norwegian Computing Center, June 1999.
- [5] G. Salton, A. Wong, and C.S. Yang, "A Vector Space Model for Automatic Indexing," Comm. ACM, vol. 18, no. 11, pp. 112-117,1975.
- [6] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- [7] U.Y. Nahm and R.J. Mooney, "A Mutually Beneficial Integration of Data Mining and Information Extraction," Proc. 17th Nat'l Conf. Artificial Intelligence (AAAI '00), pp. 627-632, 2000.
- [8] 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.
- [9] 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.
- [10] T. Hofmann, "The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data," Proc. 16th Int'l Joint Conf. Artificial Intelligence (IJCAI '99), pp. 682-687, 1999.