International Journal of Engineering & Technology, 7 (4) (2018) 4678-4682

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet. v7i4.15372
Research paper

A new approach to represent textual documents using CVSM

Dr. A. Brahmananda Reddy [1] *, Dr. Y. Sagar [2], Dr. P. Subhash [2]

[1] Associate Professor, Department of CSE, VNRVJIET, Hyderabad
[2] Associate Professor, Department of CSE, VNRVJIET
**Corresponding author E-mail: brahmanandareddy_a@vnrvjiet.in*

Abstract

Due to advancements in technology, a vast amount of data is produced which is generally in the form of unstructured data. This is where text mining finds its value to discover and retrieve useful information. Text mining is a process of seeking or extracting high quality information. Generally, in text mining, Vector Space Model (VSM) is used which transforms unstructured data to structured data by the use of traditional keyword based approach. One of the problems with this approach is that if a user puts a query, the set of documents are retrieved which match the keywords in the query. To overcome this, a Conceptual Vector Space Model (CVSM) is described in this paper which helps to categorize different documents with the same content which may use different vocabulary. The Conceptual Vector Space Model is implemented with the help of WordNet, Natural Language ToolKit (NLTK).Clustering algorithms are applied on it to form clusters based on concepts.

Keywords: *Text Mining; Vector Space Model; Conceptual Vector Space Model; Wordnet; NLTK; Clustering.*

1. Introduction

Text mining, as the name refers is the process of mining the textual resources which are generally in unstructured form to extract the significant knowledge and information [1]. Text Mining is believed to have high commercial potential value as most of the information i.e., more than 80% is in the text form.
The textual data is obtained from various information repositories. Pre-processing techniques [2] include various feature extraction methods which are applied to the textual data. Pre-processing task transforms the raw, unstructured data from textual data sources into a structured intermediate format. Knowledge discovery component generally used to discover valuable information from the intermediate form of textual data. Visualization component includes the Graphical User Interface for browsing facility and tools for creating and viewing patterns.

Fig. 1: Generic Text Mining Framework.

Text retrieval techniques must be effective and efficient in managing the increasing size of textual information. Most of the existing text retrieval techniques rely on indexing keywords which are widely used by the commercial systems. One such model is VSM, which uses vectors as the index terms that correspond to documents and queries. It is used for ranking of relevant documents, information extraction, evaluating search engines and indexing.
Unfortunately, keywords do not adequately capture the content of documents. This is where the WordNet[4] tool is useful for computational linguistics and natural language processing. WordNet is a lexical database of English language. Synsets, which are set of synonyms, are interlinked by means of conceptual semantics and lexical relations.
Natural Language ToolKit (NLTK)[5] is a massive toolkit used for building programs for text analysis. NLTK is the automatic or semi-automatic processing of human language. NLTK has been aided with text processing libraries for tokenization, stop word removal, stemming, classification and many more.

2. Pre-processing

Pre-processing in text mining is one of most important step as it is used for analysis of text and the processed tokens, which are the primary units that are passed to other phases of analysis. Pre-processing is utilized to limit the indexing size of the textual documents by stopword removal procedure as stopwords represent about 20-30% of the textual document. It is also used to enhance the performance of the Information Retrieval systems.
Generally, pre-processing consists of Tokenization and Stop word removal [6].

2.1. Tokenization

- Tokenization is the process of dividing the text into a set of meaningful pieces. These pieces are called as tokens.

Example: Actions speak louder than words.
In this example, the content is divided into the following meaningful tokens: Actions, speak, louder, than, words.

2.2. Stop word removal

- In Stop word removal method, unnecessary words like is, are and, etc., are removed from the text document.

Example: Actions speak louder than words.

In this Example, it removes the unnecessary words i.e., 'than' and displays action speak louder words.

3. Existing system - vector space model (VSM)

VSM is an algebraic model for representing text documents as vectors containing the frequencies of individual terms or identifiers [7]. Each document from the corpus is represented as a multi-dimensional vector. The number of unique terms in the corpus determines the dimension of the vector space.
Vector elements are the weights associated with individual terms i.e., the frequency of each term in every document. These weights reflect the relevancy of the corresponding terms in the given corpus. If a corpus consists of n terms (t_i , i=1 to n), document d from that corpus would be represented by the vector: d= {W_1, W_2,..,W_n}, where W_i represents the weights associated with termst. In VSM, corpus is represented as Term Document Matrix (TDM), i.e., an m x n matrix with following features: Rows (i =1, m) are the terms from the corpus.Columns (j=1, n) are the documents from the corpus.
Fig No.2: Traditional Vector Space Model
Cell [i, j] stores the weight of the term i in the context of the document j.
Documents from the corpus need to be preprocessed before creating the TDM matrix. The set of words to those that are expected to be the most relevant for the given corpus are to be reduced.

3.1. Computing term weights

The approaches for determining the term weights include [3]:

- Term Frequency(TF)
- TF-IDF

3.1.1. Term frequency (TF)

Term Frequency (TF) is a measure which represents the count of a term in a document, i.e., the number of times the term t appears in document d.
If the term frequency of a term is high compared to others in the document, then, it implies that the term is more important than others in the document.

TF (t) = count (t, d)

t – term in a document

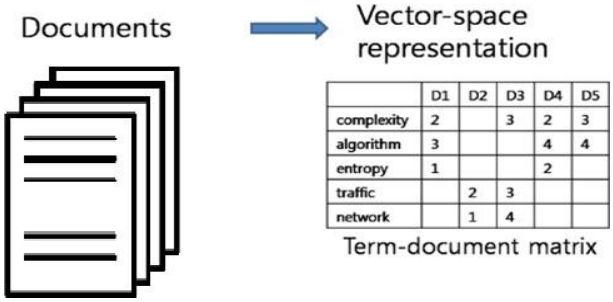
d – an individual document count (t, d) – the number of times term t appeared in the document d.

3.1.2. TF-IDF: inverse document frequency (IDF)

The idea here is to assign higher weights to unusual terms, i.e., to terms that are not so common in the corpus. It is a measure of how important the word is in the document, i.e., whether it should be considered during document retrieval or not.
It is computed in the following way:

IDF (t) = 1 + log (N / df (t))

N – Number of documents in the corpus
df (t) – number of documents with the term t



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

This is the most frequently used metric for computing term weights in a VSM.
General formula for computing TF-IDF: TF-IDF (t) = TF (t) x IDF (t)
The above formula can be written as: TF-IDF (t) = TF (t) * log (N / df (t))

3.2. Limitations of VSM

The VSM consists of the following limitations:

- High dimensionality: As the documents size increases, the dimensionality of the matrix also increases. Thus, resulting in poor processing and incorrect results.
- Complexity in semantics: Documents with different term vocabulary but similar meaning won't be related.

4. Proposed system – conceptual vector space model (CVSM)

One of the major problem in information retrieval is word matching or vocabulary problem i.e., the documents which match the terms in the query are retrieved; not the documents which deal with the same content but use different words. This simple word matching will probably miss some relevant documents just because they do not match to the terms in the query.
Here, concepts are retrieved for each pre-processed token i.e., a semantic-based approach is adapted for defining the semantics of information. A concept refers to a set of synonyms for a pre-processed token. Concept importance shows how important a concept is in Information
Retrieval. These concepts are used for relevant document retrieval.

4.1. Tokenization

NLTK can be used to tokenize given text into words. The word_tokenize function from nltk.tokenize module is used for this purpose.
From nltk.tokenize import word_tokenize
word_tokenize('Computer Processor.')
Output: ['Computer', 'Processor', '.']

4.2. Stop words removal

Stop words can be removed easily using NLTK
To check the list of stop words you can type the following commands in the python shell.

Import nltk from nltk.corpus import stopwords
Print (set (stopwords.words('english')))

The output contains all the stop words that have been incorporated in the stop words package of NLTK.
The words which are not necessary in the documents can be added to the stop words list using stopwords. append () function.

4.3. Generating concepts

Firstly, the textual documents are given as input to the NLTK tool, where it performs the above preprocessing functions. The result is the filtered text file which contains tokens which are important in the document. Then the WordNet is connected to the NLTK. The above generated filtered text file is connected to the WordNet and is processed to generate synsets for each word in the file i.e., for each token, a synset is generated. A concept refers to the set of synonyms for a preprocessed token.
To link the text documents to the WordNet tool [10], the following command is used: from nltk.corpus import wordnet

4.4. Generating CVSM matrix

A collection of n documents and m concepts with their respective weights can be represented as a CVSM matrix.

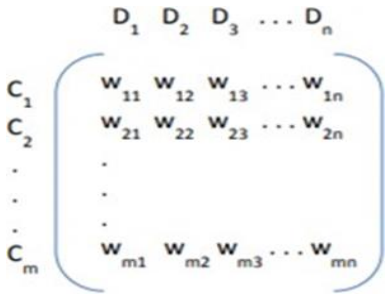


Fig. 3: Representation of a CVSM Matrix

Here, D_i to D_n = Documents i to n C_i to C_m = Concepts i to m
W_{ij} = weight of the concept C_i in the document D_j
The weight (w_{ij}) of each concept in the document can be calculated by knowing the frequency of the token and their concepts i.e., synonyms.
The formula is:

$W_{ij}(Ci,Dj) = (f_1+f_2*0.5)/(t+c)$ Here,

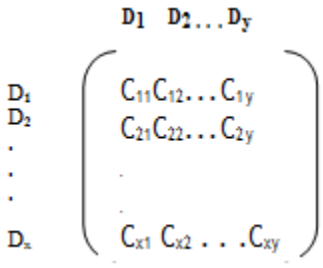
f_1 = frequency (count) of token. f_2=sum of frequencies of all concepts.
t=1, if the token is present in the document.
c = number of unique concepts that match with the words in the document.

4.5. Generating document-document matrix

The similarity between every document pair is calculated as a basis for determining the clusters. A similarity measure calculates the similarity between two documents, thus, determining the relationship between the documents. Here, the following similarity measure is applied [8].

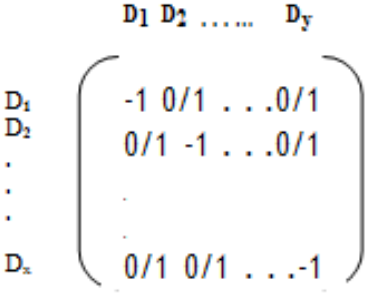
SIM (Documentx, Documenty) = ∑ (Conceptk, x) * (Conceptk, y)

Where, k is summed across the set of all concepts.
The above similarity measure takes the two rows of the two documents being measured, multiplies their corresponding values for individual documents and then sums up all those products. The results are placed in an "m" by "m" matrix, called a Document-Document Matrix, where "m" is the number of rows (documents) in the original matrix. The matrix generated from this formula is symmetric. Using the CVSM generated, the Document- Document Matrix produced is shown below.



4.6. Generating document-relationship matrix

After the Document-Document matrix has been generated, a threshold value must be chosen which determines if the two documents are similar enough such that they can be placed in the same cluster. Thus, two documents are considered to be similar if the similarity value between them is greater than or equal to the chosen threshold value. This threshold value is used to generate a binary matrix from the Document-Document matrix, called Document-Relationship matrix [8], which determines which pair of documents are similar and those which are not similar. An element '1' in the Document-Relationship matrix implies that the corresponding row and column documents are similar enough to be in the same class.



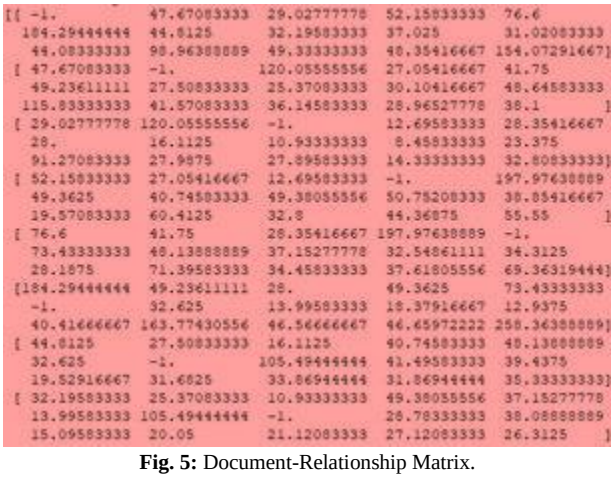


Fig. 5: Document-Relationship Matrix.

After generating the Document-Relationship matrix, the next step is to generate the clusters. Here, it is to be determined when two documents should be in the same cluster and when they should be in different clusters. For this purpose, there are several algorithms available. Here, Cliques algorithm is implemented.

5. Cliques clustering

Generally, clustering is a process of dividing a set of data into several groups, each group consisting of similar types of data objects.
The primary criteria for Cliques [8] is that, all items in a cluster must be within the threshold of all other items in that cluster, i.e., for an item to be in a cluster, it should be within the threshold of

all other items in that cluster. The Cliques clustering algorithm can be summarized as below:

1) SET item =1
2) Select doc (item) and store that in a newclass
3) Begin with doc (j), { where, i = j = item +1}
4) If doc (j) is within the threshold value, thenvalidate Else, j = j +1
5) If k > D {number ofdocs},

Then, i = i + 1

if i = D then go to Line 6, Else,

j = i

Create a new class with doc (item) in it go to Line 4
Else, go to Line 4

1) If current class has only doc(item) in it, thendelete thatclass

Else, item = item + 1

2) If item = D + 1 then go to Line8 Else go to Line 2
3) Delete duplicate classes and classes which are subsets of other existingclasses.

6. Results

The results generated by all the above-mentioned methods applied on a sample data are shown below.

6.1. Conceptual vector space model

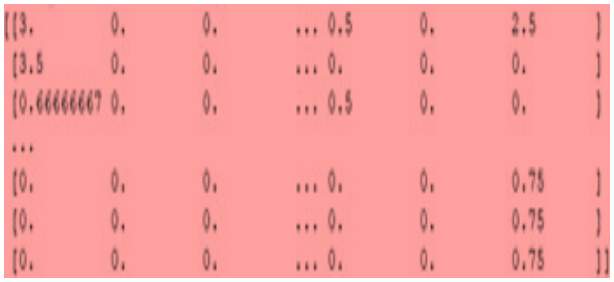


Fig. 6: Example of CVSM.

A CVSM is created, where each element represents the average weight of that concept in the respective document. The average weight includes the frequency of concept and the sum of individual frequencies of all synonyms of that concept divided by the total number of matches.

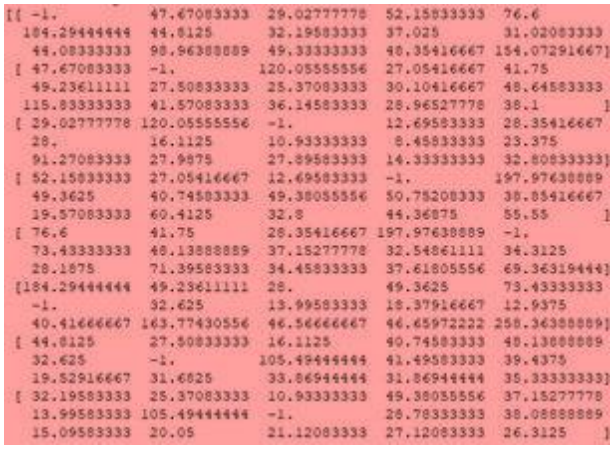


Fig. 7: Example of Document-Document Matrix.

6.2. Document-document matrix

Here, an m x m symmetric matrix is created, where; m represents the number of documents. Here, two rows of the two documents being analyzed are multiplied (separately for individual concepts) and those values are summed together.

6.3. Document-relationship matrix

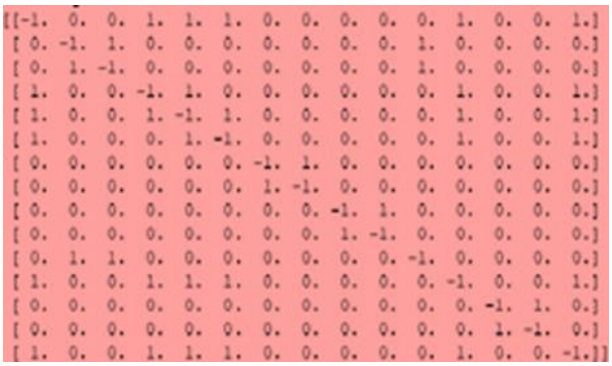


Fig. 8: Example of Document-Relationship Matrix.

The above generated Document-Document Matrix is converted into a binary matrix by initially choosing a threshold value and replacing each element with '1' if its value is greater than or equal to the threshold chosen, and with '0' otherwise. The threshold value can be chosen by calculating the average of all the values in the Document- Document Matrix.

6.4. Cliques clustering

The Cliques clustering is applied on the Document- Relationship Matrix to generate clusters. The following clusters are generated for the above Document- Relationship Matrix.

7. Result analysis

In CVSM, the documents are clustered on the basis of concepts. In VSM, the documents are clustered based on keywords.

$$\text{Precesion} = \frac{\text{Relavent Documents Retrieved}}{\text{Total Number of Documents Retrieved}}$$

$$\text{Recall} = \frac{\text{Relavent Documents Retrieved}}{\text{Total Number of Relevent Documents}}$$

$$\text{F} - \text{Measure} = \frac{2 * \text{Precesion} * \text{Recall}}{(\text{Precesion} + \text{Recall})}$$

	Precision	Recall	F-Measure
CVSM	0.615	0.533	0.570
VSM	0.619	0.520	0.565

For the sample data set taken, F-Measure for CVSM is greater than the F-Measure for VSM.

8. Conclusion and future work

On viewing the limitations of VSM, a semantic based approach is adapted for defining the semantic information. This project is incorporating semantic approach for transforming unstructured data into conceptual data by the usage of WordNet, from which concepts are generated for each processed token. This process was done on multiple documents and a matrix is generated which shows the average weight of each concept array in every document. Here, the average weight is calculated using the formula: $W_{ij} (C_i , D_j) = (f_1 + f_2 * 0.5) / (t + c)$. A similarity measure is applied to determine the similarity between documents and then, a clustering technique is applied to retrieve relevant clusters. Based on the results generated, it is proved that the Conceptual Vector Space Model retrieves more relevant and accurate documents as compared to the Vector Space Model.

The goal of information retrieval is to find relevant information related to the query fired by the user. The future scope for this project aims to assist on this task by presenting a new approach for retrieving documents based on a search by concept method, i.e., documents containing the query terms and those containing the synonyms of those query terms are also retrieved. Thus, a search engine will be created which accepts a query from the user and retrieves relevant documents based on a semanticsearch.

References

[1] Dr. G. Rasitha Banu, VK Chitra, A Survey of Text Mining Concepts, semanticsscholar.org, April 2015.

[2] Niladri Biswas, Text Mining and its Business Applications, September 2014.

[3] A. Brahmananda Reddy, A. Govardhan "Integrated Feature Selection Methods for Text Document Clustering", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.81 (2015), PP: 153-158, Research India Publications.

[4] https://wordnet.princeton.edu/.

[5] https://www.nltk.org/.

[6] Dr.S.Kannan, VairaprakashGurusamy, Preprocessing Techniques for Text Mining, Conference Paper, March 201.

[7] E. E. Ogheneovo, R. B. Japheth, Application of Vector Space Model to Query Ranking and Information Retrieval, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 6, Issue 5, May 2016.

[8] Gerald J. Kowalski, Mark T. Maybury, Information Storage and Retrieval Systems, Theory and Implementation, 2006.

[9] Brahmananda Reddy; A. Govardhan , A novel approach for similarity and indexing-based ontology for semantic web educational system, International Journal of Intelligent Engineering Informatics (IJIEI), Vol. 4 No.2, 2016. https://doi.org/10.1504/IJIEI.2016.076698.

[10] Dr. S. Vijayarani, Ms. J. Ilamathi, Ms. Nithya3, Preprocessing Techniques for Text Mining - An Overview, International Journal of Computer Science & Communication Networks, Vol 5(1), 7-16.

[11] Vaibhav Kant Singh, Vinay Kumar Singh, Vector Space Model: An Information Retrieval System, International Journal of Advanced Engineering Research and Studies, 2015.

[12] http://www.nltk.org/howto/wordnet.html.