Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 30-36, 2016
ISSN 1990-9233;
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.TAET23316

Clustering with Semantic Similarity for Text Mining

M.K. Vijaymeena and K. Kavitha

CSE, Nandha Engineering College, Erode, India

Abstract: Text processing plays a significant role in information retrieval, data mining and web search. Semantic similarity is used to analyse the relationship between word-pairs. The High-quality information is derived from mined text data through the devising of patterns. It is necessary to organize large amounts of unstructured text documents into a small number of conceptual clusters. Mining Text involves the pre-processing of document collections, text categorization and classification, extracting information and term from data sets. This work proposes a lexical pattern extraction technique to retrieve documents based on the user query and to find semantic relations that exist between the web documents. Coefficient measures are used to measure the similarity between the document sets. The Similarity measures are evaluated on various real world data sets for text classification and clustering problems.

Key words: Categorization · Lexical pattern · Similarity measure · Unstructured text

INTRODUCTION

Text mining (or text data mining) refers to the method of deriving high-quality information from text documents. Text mining usually involves the following steps. The method or process of structuring the text input, deriving patterns within the structured data and finally evaluation and interpretation of the output are called as text mining. 'High quality' in text mining usually refers to some combination of relevance, novelty and interestingness.

Text analysis involves retrieval of information, lexical analysis for studying word frequency, recognition of patterns, information extraction from the documents or data set, data mining techniques including visualization [1] and predictive analytics. The main goal is to turn text into Data for analysis, via application of natural language processing (NLP) and analytical methods. The web documents should be preprocessed and given as an input to the application. Coefficient measures are used to measure the similarity between the document sets. Lexical pattern extraction algorithm is used to cluster the similar documents based on the user query. It is necessary to measure the semantic relation between the documents. Text mining methods are used to derive the high-quality information and to measure the similarity between the web documents.

An important key element of text mining is its focus on the document collection. The collection of documents can be the grouping of text-based documents. Most of the text mining solutions were aimed at discovering patterns across very large document collections. The number of documents in the document collections can range from thousands to millions. The documents needed to be classified first and document collections can be either static or dynamic. Static refers to the case in which the initial complement of documents remains unchanged. Dynamically refers to the document collections which were characterized by their inclusion of updated documents over time. Figure 1.1 describes the process of text mining.

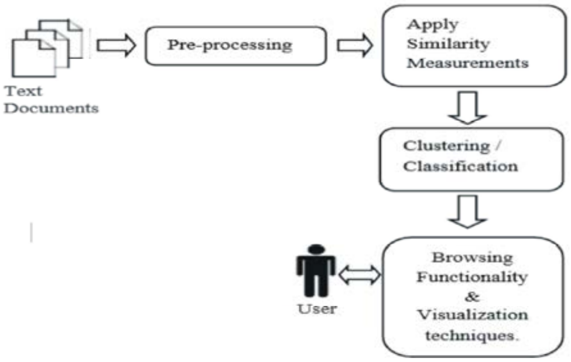
Text Documents
Pre-processing
Apply Similarity Measurements
Clustering / Classification
Browsing Functionality & Visualization techniques.
User

Fig. 1.1: Text Mining Architecture

Corresponding Author: M.K. Vijaymeena, CSE, Nandha Engineering College, Erode, India.

Literature Survey: Semantic information which is stored in ontology such as WordNet has been widely used to improve the quality of text clustering [2, 3] and it considers the semantic relationships among words [4].

Similarity Measures have been proposed to find out the common features [5] and instead of single view points, multiple view points are used to gather more information [6].

Keywords or important features was extracted from the texts and relevant information is retrieved [7]. A useful short text feature extension method [8] based on frequent term sets is also proposed to overcome the drawbacks of the vector space model on representing short text contents.

Verification techniques are effective means to reduce the number of vulnerabilities in post-release software products [9]. Visualization techniques such as VarifocalReader are used to extract information from the large set of documents and present it to users.

Expressions which are relevant are extracted using automatic keywords extraction from texts are applicable in many diverse areas such as Information Retrieval, clustering, or classification and indexing of documents.

Yung-Shen Lin, Jung-Yi Jiang and Shie-Jue Lee proposed a new similarity measure to Measure the similarity between a collection of documents and it is a very important operation in the text processing field. The multi-level hierarchical citation networks are captured by a process involving a Bernoulli process [10] and knowledge are discovered from scientific articles.

Ning Zhong, Yuefeng Li and Sheng-Tang Wu present an innovative pattern discovery technique [11] and it includes the pattern deploying and pattern evolving processes, discovered patterns are updated to find relevant information.

Zhiyang He, Ji Wu, Tao Li proposed LCMM to depict the multiple labelled documents and it could be used for multi-label text categorization [12]. Charu C. Aggarwal, designed an algorithm to make use of the side-information that is available with the text documents such as the links in the document, provenance information presented in the document, access behaviour of the user from logs which would be embedded into the text document [13].

Similarity Measures: Measuring the similarity between documents is one of the important operations in the text processing field. The similarity between two documents on a feature or keyword. The proposed measure takes the following three cases into account: a) The Keywords or feature appears in both documents, b) the Keywords or feature appears in only one document and c) the Keywords or feature appears in none of the documents. In the first case, the similarity between the documents increases as the difference between the two involved feature values decreases. Moreover, the contribution of the difference is normally scaled. In the second case, a fixed or default value is contributed to the similarity. Coefficient measures are statistical measures for measuring the similarity between the documents. Dice's coefficient [1] is defined as two times the number of terms which are common in the compared strings and divided by the total number of terms present in both strings.

$$S_{Dic}(\mathbf{d}_1, \mathbf{d}_2) = \frac{2\mathbf{d}_1 \cdot \mathbf{d}_2}{\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2}.$$

Euclidean distance was measured by calculating the square root of the sum of squared differences between the two vectors elements.

$$d_{Euc}(\mathbf{d}_1, \mathbf{d}_2) = [(\mathbf{d}_1 - \mathbf{d}_2) \cdot (\mathbf{d}_1 - \mathbf{d}_2)]^{1/2}$$

Jaccard similarity [2] was computed in such a way that the number of shared terms divided by the number of all unique terms present in both strings.

$$S_{EJ}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{\mathbf{d}_1 \cdot \mathbf{d}_1 + \mathbf{d}_2 \cdot \mathbf{d}_2 - \mathbf{d}_1 \cdot \mathbf{d}_2}$$

IT-Sim Measure is used to measure the document similarity.

$$S_{IT}(\mathbf{d}_1, \mathbf{d}_2) = \frac{2\sum_{w_i} \min(p_{1i}, p_{2i}) \log \pi(w_i)}{\sum_{w_i} p_{1i} \log \pi(w_i) + \sum_{w_i} p_{2i} \log \pi(w_i)},$$

where w_i denotes feature words, p_{ji} indicates the normalized value of w_i in document d_j for j = 1or j = 2, $\pi(w_i)$ is the proportion of documents in which w_i occurs.

Methodology: This Project deals with retrieving documents based on the user query and it is the most common text retrieval task. It is required to apply stemming before text documents are taken. To reduce the content size, Stop words are removed. Cosine Similarity is used to find Similarity between the documents. Lexical pattern extraction algorithm is used to find semantic relations that exist between given words. The extracted lexical patterns are clustered based on the similarity degree on given cluster. Every cluster contains one or more patterns. The pattern expresses similar semantic relations.

Figure 4.1 shows the steps involved in the generic text mining system.

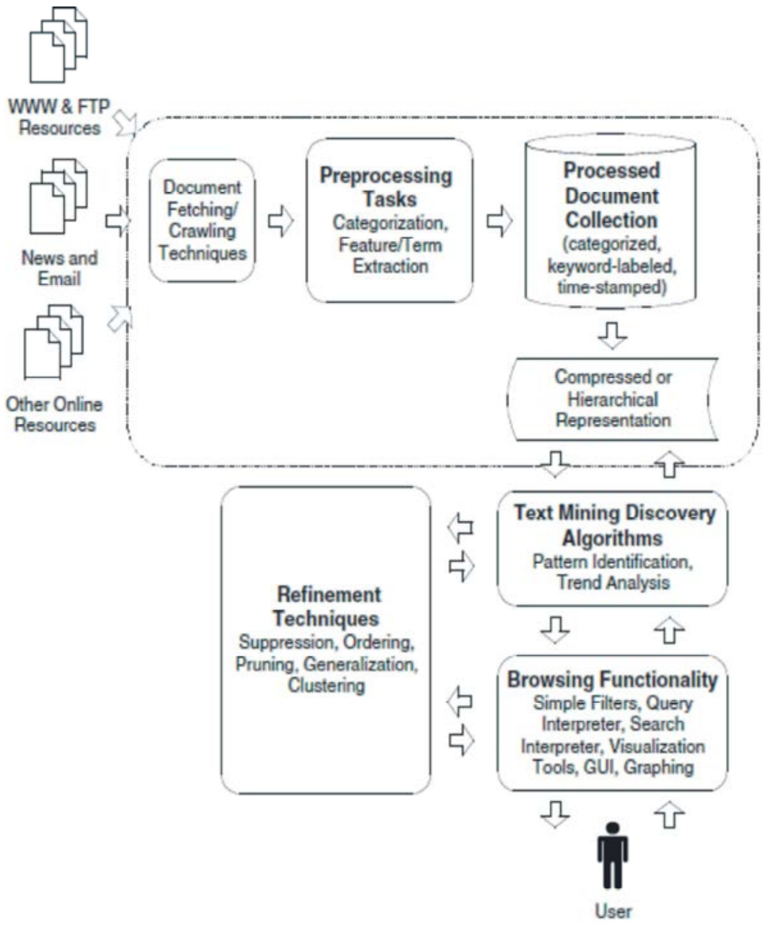
WWW & FTP Resources
News and Email
Other Online Resources
Document Fetching/ Crawling Techniques
Preprocessing Tasks
Categorization, Feature/Term Extraction
Processed Document Collection
(categorized, keyword-labeled, time-stamped)
Compressed or Hierarchical Representation
Text Mining Discovery Algorithms
Pattern Identification, Trend Analysis
Refinement Techniques
Suppression, Ordering, Pruning, Generalization, Clustering
Browsing Functionality
Simple Filters, Query Interpreter, Search Interpreter, Visualization Tools, GUI, Graphing
User

Fig. 4.1: System Architecture for Generic Text Mining System

Existing Technique: Some measures which have been popularly adopted to compute the similarity between the documents were presented in the existing system.In Single label classification experiment; the classification accuracy is the performance measure used for evaluation. In Multi-label classification approach experiment, accuracy and entropy are adopted to gauge the clustering performance. K-means clustering is used to cluster the similar groups of documents.

Several drawbacks are present in the existing system. Only text documents were taken. Web pages or XML files were not considered. Context-based knowledge or relativeness properties by modifying the text content were not considered. The existing method does not follow a sequential order of selecting clusters.

The text similarity analysis consists of the following problems.

- Relationships between the terms are not considered
- Reduction of Dimensionality is not performed
- Future weights are ignored in the similarity estimation
- Clustering accuracy is limited.

The drawbacks that were presented in the existing system are overcome in the proposed technique.

Proposed Technique: Similar documents are retrieved based on the user query. Lexical pattern extraction algorithm extracts lexical patterns from snippets retrieved from a web search engine. It is used to find semantic relations that exist between given documents. The extracted lexical patterns are clustered based on the similarity on given cluster. Each and every cluster contains patterns that express similar semantic relations.Sequential pattern clustering algorithm is used to identify semantically related patterns and to group the documents into cluster set. Figure 6.1 shows the steps involved in the proposed technique.

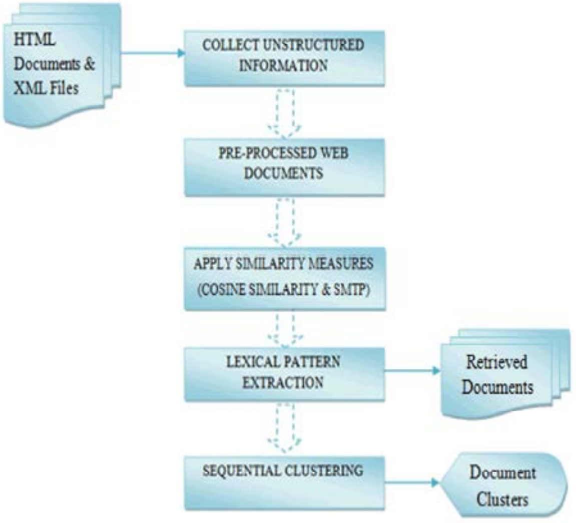
HTML Documents & XML Files
COLLECT UNSTRUCTURED INFORMATION
PRE-PROCESSED WEB DOCUMENTS
APPLY SIMILARITY MEASURES (COSINE SIMILARITY & SMTP)
LEXICAL PATTERN EXTRACTION
Retrieved Documents
SEQUENTIAL CLUSTERING
Document Clusters

Fig. 6.1: Proposed System Flow Diagram

Dataset: Documents and word pairs were obtained from Miller-Charles dataset and WordSimilarity-353.

Pre-processing: Pre-processing is an essential step to analyze the multivariate datasets before clustering or data mining. The text documents were parsed into words. Document cleaning is applied to remove stop words. Stemming process is applied to detect the inflected or derived term. Terms are updated with their feature values. Documents were pre-processed by applying stemming, stop word removal and synonym word replacement. Stem words were added into 'stem word' table. Stop words were added into 'stop word' table. The word and its synonym words were added into 'synonym word' table. Content length is reduced and the noises presented in the datasets were removed.

Cosine Similarity: Document similarities are measured based on the contents which were overlapped between them. It is possible to process those documents for information extraction from documents, clustering, classification and search applications. Let us consider d_1 and d_2 are document one and two respectively. It is required to measure the cosine angle between the documents.

Cosine similarity measures the cosine of the angle between d1 and d2 as follows:

$$S_{Cos}(\mathbf{d}_1, \mathbf{d}_2) = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{(\mathbf{d}_1 \cdot \mathbf{d}_1)^{1/2} (\mathbf{d}_2 \cdot \mathbf{d}_2)^{1/2}}.$$

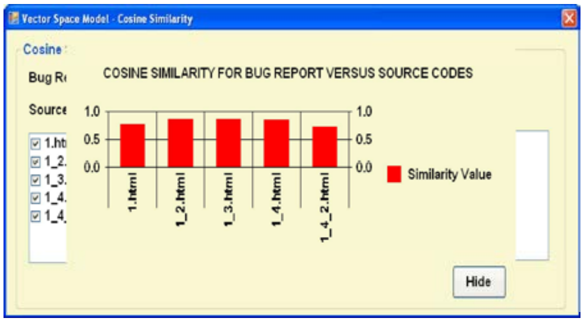

Fig. 6.2: Cosine Similarity Value

Figure 6.2 shows that if the similarity value is greater than 0.5, similarity degree increases else similarity degree decreases. If the similarity value is 1, then it shows that it was an exact match.

Similarity Measure for Processing: The similarity between two documents is found out by considering the average score of the features occurring in at least one of the two documents. A similarity measure is implemented in this module, called SMTP (Similarity Measure for Text Processing), for two documents $d_1 = < d_{11}, d_{12},..., d_{1m} >$ and $d_2 = < d_{21}, d_{22},..., d_{2m} >$. Following are the properties of the proposed similarity measures.

- The presence or absence of a feature is more essential than the difference between the two values associated with a present feature.
- The similarity degree should increase when the difference between two non-zero values of a specific feature decreases.
- The similarity degree should decrease when the number of presence-absence features increases.
- The similarity measure should be symmetric.
- Two documents are least similar to each other if none of the features are presented.

A Function F is defined as follows:

$$F(\mathbf{d}_1, \mathbf{d}_2) = \frac{\sum_{j=1}^{m} N_*(d_{1j}, d_{2j})}{\sum_{j=1}^{m} N_\cup(d_{1j}, d_{2j})}$$

where

$$N_*(d_{1j}, d_{2j}) = \begin{cases} 0.5\left(1 + \exp\left\{-\left(\frac{d_{1j}-d_{2j}}{\sigma_j}\right)^2\right\}\right), \\ \qquad \text{if } d_{1j}d_{2j} > 0 \\ 0, \ \text{if } d_{1j} = 0 \text{ and } d_{2j} = 0 \\ -\lambda, \ \text{otherwise,} \end{cases}$$

$$N_\cup(d_{1j}, d_{2j}) = \begin{cases} 0, \ \text{if } d_{1j} = 0 \text{ and } d_{2j} = 0 \\ 1, \ \text{otherwise.} \end{cases}$$

SMTP expression is

$$S_{SMTP}(\mathbf{d}_1, \mathbf{d}_2) = \frac{F(\mathbf{d}_1, \mathbf{d}_2) + \lambda}{1 + \lambda}.$$

Similarity Measure for Between Two Document Groups: The Similarity between two document sets is designed to calculate an average score of the features occurring in the two sets. Let G_1 and G_2 be two document sets containing q_1 and q_2 documents. The function F between G_1 and G_2 is defined to be

$$F(G_1, G_2) = \frac{\sum_{i=1}^{q_1} \sum_{j=1}^{q_2} \sum_{k=1}^{m} N_* \left(d_{ik}^1, d_{jk}^2\right)}{\sum_{i=1}^{q_1} \sum_{j=1}^{q_2} \sum_{k=1}^{m} N_\cup \left(d_{ik}^1, d_{jk}^2\right)} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{q_1} \sum_{j=1}^{q_2} N_* \left(d_{ik}^1, d_{jk}^2\right)}{\sum_{k=1}^{m} \sum_{i=1}^{q_1} \sum_{j=1}^{q_2} N_\cup \left(d_{ik}^1, d_{jk}^2\right)}$$

Similarity measure, SMTP is

$$S_{SMTP}(G_1, G_2) = \frac{F(G_1, G_2) + \lambda}{1 + \lambda}.$$

λ is the tuning parameter used for optimizing the system and based on the assumption, value of λ is 1. A low value of λ is directly proportional to high efficiency.

Lexical Pattern Extraction: Lexical pattern extraction algorithm uses snippets retrieved from a web search engine. The search pattern were entered in which the first word and last word was taken. In the web pages, the phrase is checked such that the pattern is first word, any number of words and the last word. During the pattern extraction, the skip count number of words can be discarded in the phrase. The search pattern is found out from the web pages and the pages names were added to a list.

Lexical Pattern Extraction Algorithm

Input: Word Pair W

Output: Extracted Patterns and documents

```
    For each word-pair (P, Q ) □ W
    do A ← Get-Snippets (“P Q”)
    N ← null
    For each snippet a □ A
    do N ← N + Get-N-grams (a, P, Q)
    Pats ← Count-Freq (N)
    Return (Pats)
```

Sequential Clustering: A semantic relation was expressed using more than one pattern. Semantically related patterns should be grouped by using Sequential pattern clustering algorithm [14]. The count and co-occurrence of the words were obtained. The cluster can be grouped based on the threshold value entered in the textbox [15] control.

Lexical Skip Pattern Clustering Algorithm

Input: patterns A (a_1, …, a_n), threshold θ

Output: Clusters C

```
    SORT (A)
    C ← {}
    For pattern $a_i \in$ A
    do
    max ← - ∞
    $c^*$ ← null
    for cluster $c_J \in$ C do
    sim ← cosine ($a_i$, $c_J$)
    if sim > max then
    max ← sim
    c ← $c_J$
    end if end for
     if max > $\theta$ then
    c* ← c* $\oplus$ $a_i$
    else
    C ← C $\cup$ {$a_i$}
    end if
    end for
    return C
```

Experimental Resuts: The following Table 1 and Fig 7.2 show experimental result for existing system analysis. The table contains word pair, word Jaccard value, word overlap values, word dice values, word PMI values and its PMI max values details were shown. The pair word count details are measure the correlation coefficient score value in each word pair using precision and recall measure. The overall word pair coefficient values are Jaccard value 0.584, the overlap value is 0.875, dice values is 0.695, PMI values are 2.846 and PMI max values are 4.025.

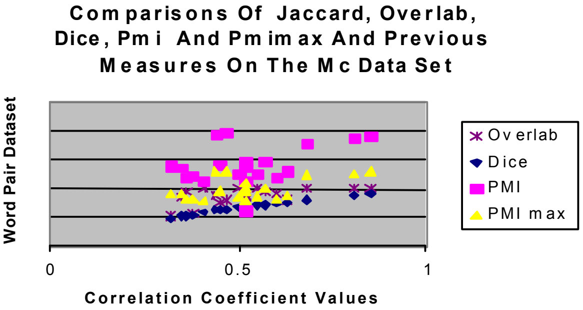

Fig. 7.1: Comparisons of Values Mc Data Set

Table 1: Comparison of Existing Word pair Coefficient and lexical pattern Clustering Algorithm

List of Words	Existing Word Pair Coefficient	Skip Lexical Clustering Algorithm
Pain	0.8235	0.654
Cord	0.8153	0.663
Smile	0.8003	0.351
String	0.8213	0.320
Noon	0.8441	0.322
Blue	0.8521	0.567
Car	0.8	0.775
Automobile	0.8235	0.234
Forest	0.8235	0.789
Coast	0.8235	0.586

Table 2: Cluster Size in Lexical Pattern Clustering

S.No.	Threshold values	Cluster Size
1	0.5	1
2	0.6	2
3	0.8	2
4	0.9	4
5	1	5

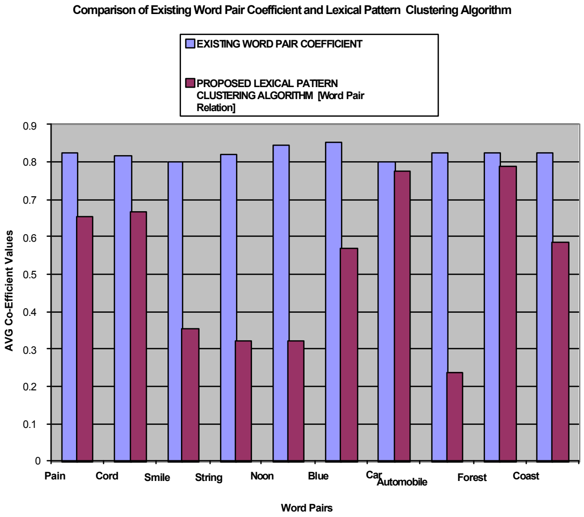

Fig. 7.2: Comparisons of Values Mc Data Set

Figure 7.2 shows the comparison between Existing Coefficient and Skip Lexical Pattern Clustering Algorithm on word pairs. Table 2 shows the cluster size in lexical pattern clustering and the size of threshold values is directly proportional to the cluster size. Figure 7.3 plots average similarity between features vector and all word pairs which are synonymous for different values of θ. From the graph, we could see that average similarity initially increases when θ is increased. To reduce the feature vectors sparseness, semantically related patterns were clustered. Average similarity values are stable when the θ values ranges between 0.5 and 0.7. Furthermore, increasing θ beyond 0.7causes a rapid drop of average similarity. HTML tags are removed from bug report and source code entity file before taken for similarity identification.

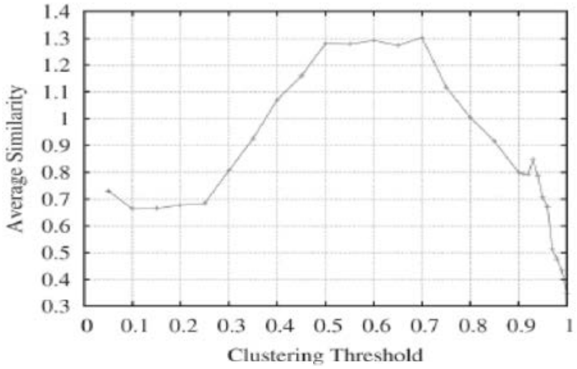

Fig. 7.3: Average similarities versus clustering threshold

CONCLUSION

This project presents an enhanced similarity measure between the set of documents. Accuracy and Similarity values are used to evaluate the system performance. Several desirable properties were embedded in this measure. Pre-processing of the dataset is done and documents were obtained. The proposed methodology uses lexical pattern extraction to extract numerous semantic relations that exist between the words should be done. Sequential pattern clustering needs to identify different lexical patterns that describe the same semantic relation. In future, searching and comparing the video content based on the exact semantic words could be implemented. If the project is developed as web service, it can be accessed from anywhere. The project can be further developed by working in different operating system independently.

REFERENCES

1. Steffen Koch, 2014. Member, IEEE, Markus John, Michael Worner andreas Muller and Thomas Ertl, Member, IEEE VarifocalReader - In-Depth Visual Analysis of Large Text Documents, 20(12): 1723-1732.
2. Liang, J.G., X.F. Zhou, P. Liu, L. Guo and S. Bai, 2013. An EMM-based Approach for Text Classification, Procedia Computer Science 17: 506-513.
3. Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, Xianyu Bao, 2015. A semantic approach for text clustering using WordNet and lexical chains, Expert Systems with Applications, 42: 2264-2275.

4. Dnyanesh G. Rajpathak, 2014. Member, IEEE and Satnam Singh, Senior Member, IEEE An Ontology-Based Text Mining Method to Develop D-Matrix from Unstructured Text, IEEE Transactions On Systems, Man and Cybernetics: Systems, 44: 966-977.
5. Yung-Shen Lin, Jung-Yi Jiang and Shie-Jue Lee, 2014. Member, IEEE A Similarity Measure for Text Classification and Clustering, IEEE Transactions On Knowledge And Data Engineering, 26(7): 1041-4347.
6. Andrew Skabar, 2013. Member, IEEE and Khaled Abdalgader "Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm, IEEE Transactions On Knowledge And Data Engineering, 25(1): 62-75.
7. Joa Ventura and Joaquim Silva, 2012. Mining Concepts from text, Procedia Computer Science, 9: 27-36.
8. Yuan Man, 2014. Feature extension for Text Categorization using Frequent term set, Procedia Computer Science, 31: 663-670.
9. Riccardo Scandariato, James Walden, Aram Hovsepyan and Wouter Joosen, 2014. Predicting Vulnerable Software Components via Text Mining, IEEE Transactions On Software Engineering, 40(10): 993-1006.
10. Zhen Guo, Zhongfei (Mark) Zhang, Shenghuo Zhu, Yun Chi and Yihong Gong, 2014. A Two-Level Topic Model Towards Knowledge Discovery from Citation Networks, IEEE Transactions On Knowledge And Data Engineering, 26(4): 780-794.
11. Ning Zhong, Yuefeng Li and Sheng-Tang Wu, 2012. Effective Pattern Discovery for Text Mining, IEEE Transactions On Knowledge And Data Engineering, 24(1): 30-44.
12. Zhiyang He, Ji Wu and Tao Li, 2014. Label Correlation Mixture Model: A Supervised, Generative Approach to Multi-label Spoken Document Categorization, DOI10.1109/ TETC.2014.2377559, pp: 1-11.
13. Charu C. Aggarwal, 2014. Fellow, IEEE Yuchen Zhao and Philip S. Yu, Fellow, IEEE On the Use of Side Information for Mining Text Data, IEEE Transactions On Knowledge And Data Engineering, 26(6): 1415-1429.
14. Duc Thang Nguyen, Lihui Chen, Senior Member, IEEE and Chee Keong Chan, 2012. Clustering with Multiviewpoint-Based Similarity Measure, IEEE Transactions On Knowledge And Data Engineering, 24(6): 988-1001.
15. Jehoshua Eliashberg, Sam K. Hui and Z. John Zhang, 2014. Assessing Box Office Performance Using Movie Scripts: A Kernel-Based Approach, IEEE Transactions On Knowledge And Data Engineering, 26(11): 2639-2648.