AENSI Journals

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Efficient Text Clustering Using C^{2}S Classifier

[1]Subbaiah Shanmugasundaram and [2]Chandrasekar Chelliah

[1]*Department of Computer Applications(MCA), K. S. Rangasamy College of Technology, Tiruchengode 637 215 , TamilNadu, India.*
[2] *Department of Computer Science, Periyar University, Salem 636 011, TamilNadu, India.*

ARTICLE INFO
Article history:
Received 14 October 2013
Received in revised form 21 November 2013
Accepted 22 November 2013
Available online 15 December 2013

Key words:
Clustering, Text Mining, Synset, NLP

ABSTRACT
Text clustering methods have various deficiencies like overlap, inefficiency, speed and time. Due to the increase in number of concepts and number of documents, clustering becomes more difficult one where reducing overlap, time and increasing speed and efficiency of clustering. We propose a new technique using various measures like Conceptual Strength Measure, Conceptual Depth Measure, and Semantic Similarity Measure to identify the cluster. Each text document is preprocessed to identify monogram and bigrams by removing stop words and performing stemming process and we used stanford part of speech tagger to identify nouns. At the second stage we compute all the four measures, we used ODP taxonomy and wordnet to collect the terms related to each cluster , concepts , root class and subclass and pointers from wordnet. Finally we compute a combined weight to identify the cluster of the text document.

To Cite This Article: Subbaiah Shanmugasundaram, Chandrasekar Chelliah., Efficient Text Clustering Using C^{2}S Classifier. ***Aust. J. Basic & Appl. Sci.,*** *7(13): 107-113, 2013*

INTRODUCTION

Text clustering has been reviewed of past 25 years in the area of information technology, data mining. The application of text clustering has impact in various domains and application. Whenever the volume of document increases, the process of mining or extracting information becomes complicated. In order to extract or mine exact document or knowledge is completely depend on the kind of clustering methodology used.

Clustering is a way of organizing or indexing the text document, so that the document can be retrieved easier at later stage. The text document contains several paragraphs, each paragraph may contain many statements and a statement is combination of terms. For clustering purpose each documents has to be analyzed and identified about the concept what it is talking about. For example if we give 100 documents, each documents topic has to be identified, so that we can index the document into a cluster. Some of the document may discuss about data mining and few of them may speak about image processing and so on. The problem here is what we do when the volume of documents increases; this is where we start thinking about the clustering algorithms which works on computerized manner.

When we computerize the process of text mining or text clustering, the time complexity and overlap of clustering has to be taken care of. Sometimes a single document may be indexed into different cluster or indexed into a wrong category. So that, selecting a clustering algorithm have few key points to reduce the time complexity and to increase the efficiency of the clustering algorithm.

We propose a new text clustering algorithm which is based on the conceptual and semantic measures. We compute the conceptual measures and identify the cluster to which the document is belongs to. The CCBS used standford part of speech tagger and word net and ODP taxonomy to compute the conceptual measures. We used pos tagger to identify the key terms present in the document and wordnet is used to identify synset pointers which specifies the related terms. ODP taxonomy is an open source hierarchical dictionary which contains set of terms below a single category.

Background:

There are various inventions which have proposed by various research peoples. We discuss the basic methods here for the better understanding. Hierarchical clustering is proposed earlier, where the documents are clustered hierarchically, in this the time complexity is more. In order to search a document in this way of clustering the processing time is more.

K-means clustering is proposed, where the documents are indexed using the distance between the text documents and the terms in the document. Here the problem of false indexing is present and time complexity is

Corresponding Author: Subbaiah Shanmugasundaram, Department of Computer Applications (MCA), K. S. Rangasamy College of Technology, Tiruchengode 637 215, TamilNadu, India.
E-mail: subbaiah.phd123@gmail.com

also more. Frequent term based indexing is also proposed, which cluster the document only using the frequency of the terms. For each term in the document the number of occurrence is calculated and total terms are computed and finally frequency of particular term is also computed. Term Frequency TF is computed and Inverse Document Frequency IDF is calculated Based on computed values TF, IDF and Entropy weight is calculated. Based on the calculated weight the document will be indexed to a single cluster.

Frequent pattern based clustering also proposed, here we identify the set of pattern of occurrence of terms is calculated and the document is indexed.

2) A Probabilistic Analysis of the Rocchio Algorithm with tfidf for Text Categorization is proposed but lags with efficiency due to the time taken to select few terms from bag of words.

3) To reduce the processing time various dimensionality reduction techniques like Information Gain, Mutual Information, Chi-Square, Odds ratio, and so on.

4) The web document management methodology is proposed which uses selection of unigrams and bigrams for document indexing, unigrams are single nouns and bigrams are double consecutive nouns. Both unigram and bigram are used with the taxonomy to identify the class of document.

5). Term-based ontology mining methods are presented in Ontology Learning for the Semantic Web, which uses synonymy and hyponymy relation between words.

6). Apriori like algorithms are proposed for pattern mining from large set of documents, however searching interested pattern is a difficult problem

7) An Effective Hash-Based Algorithm for Mining Association Rules proposed for text categorization using association rules. The rule mining techniques are uses support and count methods to generate the association rules. Based on the generated rules the document could be indexed to a category.

8. A Two-Stage Text Mining Model for Information Filtering, is proposed, which combined term-based and pattern based approach to reduce the mismatch problem of information filtering and retrieval.

9. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments is discussed, which works based on the probability of document related to the user query or search.

10. An Effective Rule-Based Probabilistic classifier for text mining is discussed. It presents a methodology to generate positive and negative rules. Based on generated rules they calculate probability values to identify the category of document.

Proposed Method:

The proposed method has four stages like preprocessing, calculating Concept Match Measure, Concept Density Measure, Semantic Informative Measure and calculating Combined Weight, at last Clustering.

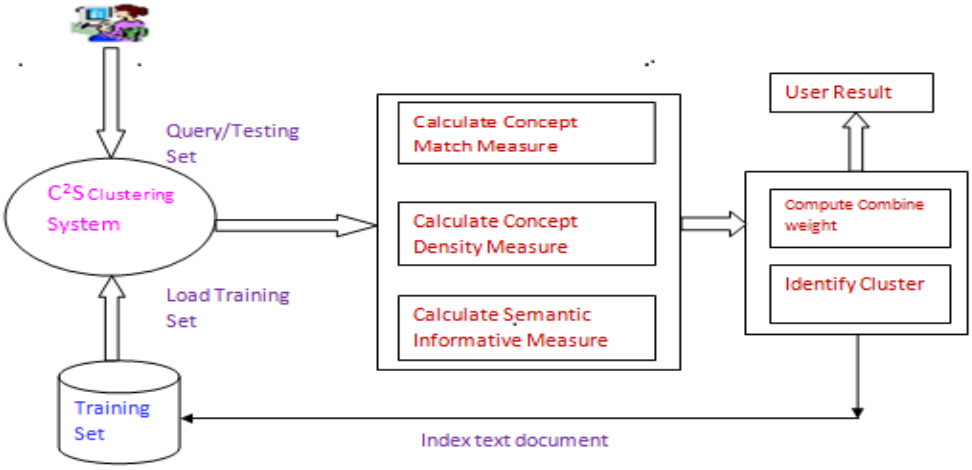


Fig. 1: System Architecture

A. Preprocessing:

At the preprocessing stage, our proposed method reads the testing documents and preprocesses the documents from the document set D_s. It reads the textual information from each document D_i from the document set $_{Ds}$ and generates a term set T_s. From the term set T_s , unnecessary words are removed as stop words and verbs also identified using standford part of speech tagger and removed from the term set. With the remaining terms in the term set T_s, stemming process is performed to get pure nouns from the terms in the term set. The selected pure nouns are used to calculate other measures to compute combined weight.

a) Preprocessing Algorithm:

Step1: Read the documents in the Training set D_s.

Step 2: For each document D_i from Document Set D_s.
Extract text content from Document D_i.
Split text into Paragraphs.
Split paragraph into statements.
Remove punctuation marks.
Split text into individual terms and collect as term set T_s.
Remove stop words from the term set T_s.
For each term in the term set T_s.
Use pos tagger to identify verb/noun.
If verb
Remove from Term set T_s.
Else
Perform stemming process.
End.
Step 3: Return Terms Set T_s.

B. Concept Match Measure (CMM):

The concept match measure is calculated follows. For a set of terms T_s given, the concepts from the taxonomy are extracted and numbers of term matches with the class lables are calculated. Wordnet and odp taxonomy is used to as corpus. Wordnet is a tool, which gives various words and synonyms for a single word and related words and odp taxonomy is a data dictionary which contains various concepts and terms in hierarchical manner.

a) Concept Match Measure (CMM) Algorithm:

Step1: For each term in the term set T_s.
For each concept in the corpus
Find exact match n or partial match m.
Compute n and compute m.
End
End
Step 2: Compute Concept Match Measure as follows
CMS $[t \in T_s] = {}^{n} \sum_{i=1} I(P_{i,t}) \times 5 \log (n + 2-i)$
Step 3: End.

Let T_s be the set of terms to be checked and P be the set of potential concept lables obtained from the corpus. And n is the number of terms collected from the corpus.

CMS $[t \in T_s] = {}^{n} \sum_{i=1} I(P_{i,t}) \times 5 \log (n + 2-i)$
$I(P_i, t) = 1$, then t contains a class with label matching P_i.
$I(P_i, t) = 0.4$, then t contains a class label with partial match.
$I(P_i, t) = 0$, then no match found.

C. Concept Depthness Measure (CDM):

The depthness measure is calculated about other concepts which are coming under a root concept. It is calculated as number of sub classes it discussed in the document, so that we call it as depthness of the concept discussed in particular document. For example if a document discusses about computer then how much it discuss about software/hardware/programming etc.

$CDM = 1/(n+m) (E_a * \mu + P_a * \beta)$
$\mu = 0.6$; $\beta = 0.4$;
Where, Let O is the set of concepts in the corpus o, and t is the set of search terms.
E_a (o, t) is the set contains the exact class match in the particular concept with the query term.
P_a (o, t) is the set contains the partial class match in the particular concept with the query term.
n = number of exact match
m = number of partial match.

a) Concept Depthness Measure (CDM) Algorithm:

Step1: For each term in the term set T_s.
For each concept in the corpus O
Find exact match n or partial match m with concept.
Compute n and compute m.
Compute E_a, P_a.

End
End
Step 2: Compute Concept Depthness Measure as follows
CDM = 1/ (n+m) (E_{a*} μ +P_a* β)
Step 3: End.

D. Semantic Informative Measure:

The semantic informative measure shows that how informative the particular document is .The semantic informative measure represented by the weighted probability of instance parent concept and instance of child concept. The semantic informative measure is calculated as follows.

SIM = 1/ (n+m) (R_{a*} μ +C_a* β)
μ = 0.6; β = 0.4;
Where, Let O is the set of concepts in the corpus o, and t is the set of search terms.
R_a (o, t) is the set contains the exact match of parent class in the particular concept with the query term.
C_a (o, t) is the set contains the exact match of child class in the particular concept with the query term.
n = number of exact match
m = number of partial match.

a) Semantic Informative Measure (SIM) Algorithm:

Step1: For each term in the term set T_s.
For each concept in the corpus O
Find exact match of parent class with the corpus R_a.
Find exact match of child class with the corpus C_a.
Compute E_a, P_a.
End
End
Step 2: Compute Semantic Informative Measure as follows
SIM = 1/ (n+m) (R_{a*} μ +C_a* β)
Step 3: End.

E. Combined Weight Calculation:

Once all those three measures been calculated, the combined score of all those analytical measure will be calculated. The following equation shows how the combined score is calculated.

Total score (d ϵ D) = $\sum_{i=1}^{3} w_i \frac{M[i]}{max_{1 \le j \le |o|} M[j]}$

Let M = { M[1] , M[2] , M[3]]}={CMM,CDM, SIM}
W_i – weight factor.
D – The set of document to cluster.

RESULTS AND DISCUSSION

The proposed system produces very good results compare to other algorithms. We used 2 million text documents and 200 concepts to cluster the document. The algorithms used 70 percent of documents for training and 30 percent as testing documents.

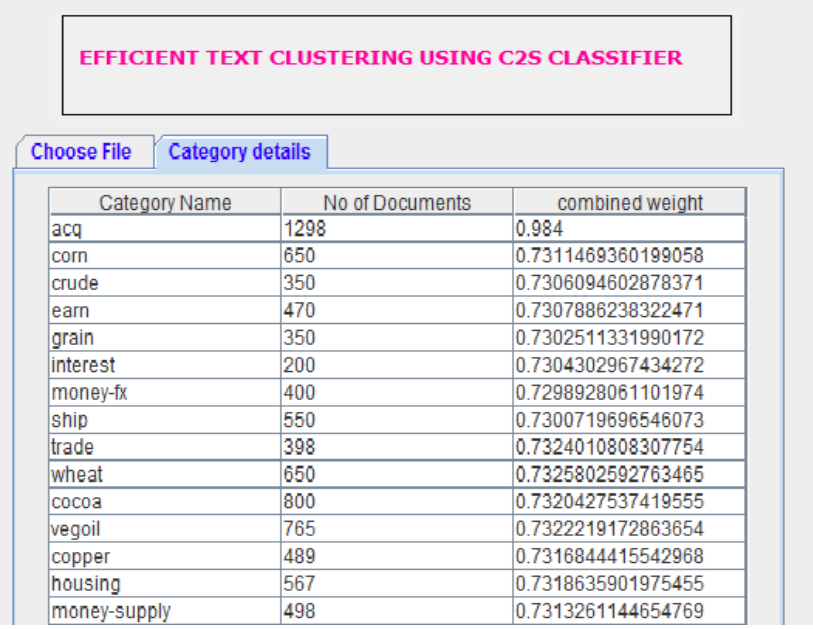
We used ten categories for testing purpose of our algorithm and the following table shows the number of documents used to evaluate the testing.

Category	No of documents
Acq	1298
Corn	650
Crude	350
Earn	470
Grain	350
Interest	200
Money-fx	400
Ship	550
Trade	398
Wheat	650
Cocoa	800
Veg-0il	765
Copper	489

Housing	567
Money-supply	798
Coffee	468
Sugar	257
Reserves	573
Ship	678
Cotton	325
Carcass	767

Fig. 1: shows the category names and number of documents used for clustering.

The Figure1 shows that number of documents used in each category for training phase. In training phase each document in all the categories are processed for clustering. A part of document in each category is used for testing purpose.



Category Name	No of Documents	combined weight
acq	1298	0.984
corn	650	0.7311469360199058
crude	350	0.7306094602878371
earn	470	0.7307886238322471
grain	350	0.7302511331990172
interest	200	0.7304302967434272
money-fx	400	0.7298928061101974
ship	550	0.7300719696546073
trade	398	0.7324010808307754
wheat	650	0.7325802592763465
cocoa	800	0.7320427537419555
vegoil	765	0.7322219172863654
copper	489	0.7316844415542968
housing	567	0.7318635901975455
money-supply	498	0.7313261144654769

Fig. 2: shows the result of generated weight value.

The Figure 2 shows the computed weight using our algorithm for a document which is given as input from the category acq. The computed weight shows that the input document has more weight for the category acq.

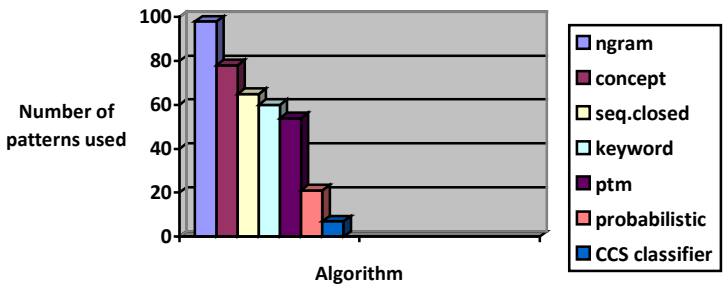


Fig. 3: Number of patterns used by different algorithms

The Figure3 shows the comparison of different algorithms and number of patterns used by them. It clearly shows that our proposed algorithm reduces the pattern size used, so that it reduces the computation time also.

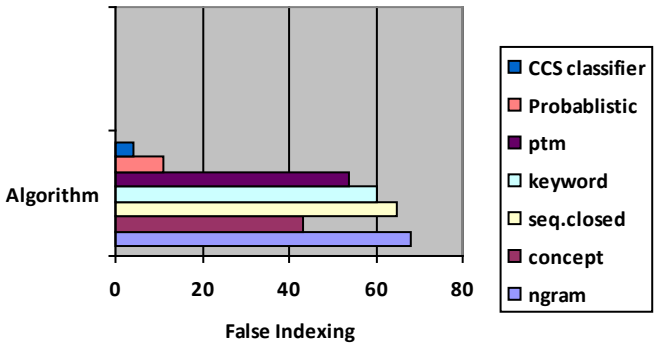


Fig. 4: shows the number of false indexing.

The Figure4 shows the variation of false indexing generated by all other algorithms. It shows that our algorithm reduces the number of false indexing, and increases the efficiency of clustering.

Conclusion:

We proposed a C^2S measure based text clustering algorithm which produces very little false indexing, overlap and produces good result. The C^2S text cluster produces good results and the measures computed are very effective and based on the computed measures combined weight is calculated. We used Reuter's data set and we split each corpus in the data set into ten categories. We used 70 percent of the corpus as training set and 30 percent as testing set. Further we can modify the measures to compute the combined weight and refine the results produced.

REFERENCES

Agrawal, R. and R. Srikant, 1994. "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), pp: 478-499.

Caropreso, M.F., S. Matwin and F. Sebastiani, 2000. "Statistical Phrases in Automated Text Categorization," Technical Report IEI-B4-07- 2000, Institution of Elaborations dell'Informazione, 2000.

Han, J. and K.C.C. Chang, 2002. "Data Mining for Web Intelligence," Computer, 35(11): 64-70.

Han, J., J. Pei and Y. Yin, 2000. "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data SIGMOD '00), pp: 1-12.

Jindal, N. and B. Liu, 2006. "Identifying Comparative Sentences in Text Documents," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06), pp: 244-251.

Joachim's, T., 1998. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. European Conf. Machine Learning (ICML '98), pp: 137-142.

Lam, W., M.E. Ruiz and P. Srinivasan, 1999. "Automatic Text Categorization and Its Application to Text Retrieval," IEEE Trans. Knowledge and Data Eng., 11(6): 865-879.

Lewis, D.D., 1992. "Feature Selection and Feature Extraction for Text Categorization," Proc. Workshop Speech and Natural Language, pp: 212-217.

Li, Y. and N. Zhong, 2003. "Interpretations of Association Rules by Granular Computing," Proc. IEEE Third Int'l Conf. Data Mining (ICDM '03), pp: 593-596.

Li, Y. and N. Zhong, 2006. "Mining Ontology for Automatically Acquiring Web User Information Needs," IEEE Trans. Knowledge and Data Eng., 18(4): 554-568.

Li, Y., X. Zhou, P. Bruza, Y. Xu and R.Y. Lau, 2008. "A Two-Stage Text Mining Model for Information Filtering," Proc. ACM 17th Conf. Information and Knowledge Management (CIKM '08), pp: 1023-1032.

Medelyan O and I.H. Witten, 2006. "Thesarus based automatic key phrase indexing", *Proc.Of the Joint Conf. on digital Libra*.

Ning Zhong, Yuefeng Li, and Sheng-Tang Wu, 2012. "Effective Pattern Discovery for Text Mining", IEEE Trans. On Knowledge and Data Engineering, 24(1): 30-44.

Odysseas Papapetrou, Wolf Siberski, and Norbert Fuhr, 2012. "Decentralized Probabilistic Text Clustering", IEEE Trans. On Knowledge and Data Engineering, 24(10): 1848-1861.

Park, J.S., M.S. Chen and P.S. Yu, 1995. "An Effective Hash-Based Algorithm for Mining Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '95), pp: 175-186.

Porter, M.F., 1980. "An Algorithm for Suffix Stripping," Program, 14(3): 130-137.

Sharma, R. and S. Raman, 2003. "Phrase-Based Text Representation for Managing the Web Document," Proc. Int'l Conf. Information Technology: Computers and Comm. (ITCC), pp: 165-169.

Shehata, S., F. Karray and M. Kamel, 2006. "Enhancing Text Clustering Using Concept-Based Mining Model," Proc. IEEE Sixth Int'l Conf.Data Mining (ICDM '06), pp: 1043-1048.

Shehata, S., F. Karray and M. Kamel, 2007. "A Concept-Based Model for Enhancing Text Categorization," Proc. 13th Int'l Conf. Knowledge Discovery and Data Mining (KDD '07), pp: 629-637.

Sparck Jones, K.., S. Walker and S.E. Robertson, 2000. "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments—Part 1," Information Processing and Management, 36(6): 779-808.

Sparck Jones, K., S. Walker and S.E. Robertson, 2000. "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments—Part 2," Information Processing and Management, 36(6): 809-840.

Srikant, R. and R. Agrawal, 1995. "Mining Generalized Association Rules," Proc. 21th Int'l Conf. Very Large Data Bases (VLDB '9 5), pp: 407-419.