ijecs
open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11267-11271

Data Analysis Using Document Clustering

Prof. Priya Thakare, Rekha Kamble Priyanka Karche, Sneha Gaikwad , Manish Khaladkar

thakarepriya1@gmail.com
rekhakamble5@gmail.com
priyanka_karche@yahoo.in
gaikwadsneha.0691@gmail.com
manishkhaladkar10@gmail.com

Sinhgad Institute of Technology and Science, Narhe, Pune 41
Savitribai Phule Pune University, INDIA.

***Abstract**:* Our proposed system uses clustering approach that organizes a large quantity of unstructured text documents into a small number of meaningful and coherent clusters. These measures are compared and analyzed in partitional clustering for text document datasets. Clustering provides extraction and fast retrieval of information or filtering. In document clustering, clustering methods can be used to automatically form group of the retrieved documents into a list of useful categories. Document clustering uses a sample documents as descriptors and performs descriptor retrieval from set of text documents. Descriptors is the sample document in reference to which clusters are formed.

Keywords – Clustering, Text mining, Data mining.

I. INTRODUCTION

Our application domain involves examining large number of files obtained by each computer. This activity increases the expert's ability of analyzing and interpreting the data. The methods for automated data analysis are widely used for machine learning and data mining are significant. Algorithms for recognition of patterns from the information present in text documents are useful. Clustering is used when there is no prior knowledge about data [2] [3]. It focuses on clustering the text documents, documents using clustering algorithms. This is done by using different combinations of parameters and different instantiations for clustering algorithms. The aim is to reduce the efforts of reading each and every document to assure its originality or relativity when a large set of documents are to be inspected.

II. RELATED WORK

There are studies regarding use of clustering algorithms in the field of Computer Forensics and other fields related to text analysis of text documents. Most of the studies describe the use of algorithms for clustering data e.g., Expectation-Maximization (EM) for unsupervised learning of Gaussian Mixture Models, K-means, Fuzzy C-means (FCM), and Self-Organizing Maps (SOM). K-means and FCM can be seen as particular cases of EM [21]. Algorithms like SOM [22] generally have inductive biases similar to K-means but are usually less computationally efficient.

In [8], SOM based algorithms were used for clustering files and making the decision- making process performed by the examiners more efficient and accurate. The files were clustered by taking into consideration their creation dates/times and their extensions. Clustered results can increase the information retrieval efficiency. It would not be necessary to review all the documents found by the user anymore.

An integrated environment for mining e-mails for forensic analysis using classification and clustering algorithms was presented in [10].The e-mails are grouped by using lexical, syntactic, structural and domain specific features in the application domain in[11]. Three clustering algorithms (K-means, Bisecting K-means and EM) were used for the e-mail clustering. The problem of clustering e-mails for forensic analysis was also addressed in [12], where a Kernel-based variant of K-means was applied. The obtained results were analysed subjectively and the authors concluded that they are interesting and useful from an investigation perspective. More recently [13], a FCM-based method for mining association rules from forensic data was described.

The literature on Computer Forensics only reports the use of algorithms that assume that the number of clusters is known and fixed already by the user. A common approach in other domains involves estimating the number of clusters from data. One induces different data partitions (with different numbers of clusters) and then assesses them with a relative validity index in order to estimate the best value for the number of clusters [2], [3], [14]. This work makes use of such method facilitating the work of the ex- pert examiner who in practice would hardly know the number of clusters *a* priori.

III. MOTIVATION

Clustering algorithms have been studied before for many years and there is huge literature on this subject. We decided to select a set of representative algorithm in order to show the potential of the proposed approach. The algoritms are partitional K-means ,Improved K-means, Porter stemmer algorithm. This algorithm were executed with different combinations of parameters and results. In order to make the comparative analysis of the algorithms more effective, two relative validity indexes have been used to estimate the number of clusters automatically from data.

Admin can have access to all functions such as uploading the documents, all pre-processing steps, display the scores of documents and clustering, at last the result is shown. Pre-processing steps involved parsing, filtering, stemming, calculation of TF (Term Frequency) – IDF (Inverse Document Frequency).

Parsing means making various tokens. Stemming means finding root word from whole word. For example: do, doing, does, done these are given words. Root word for these words is go. As the term implies, TF-IDF calculates values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents in which the word is present. Words with high TF-IDF numbers imply a strong relationship with the document in which they appear, suggesting that if that word were to be present in a query, the document could be of interest to the user.

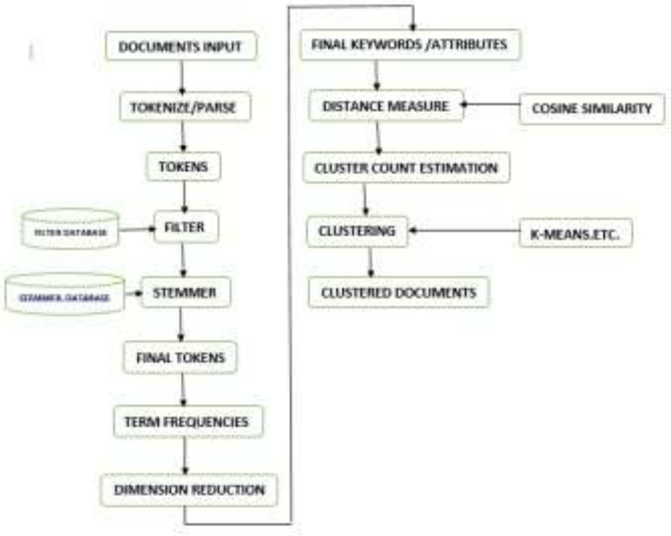


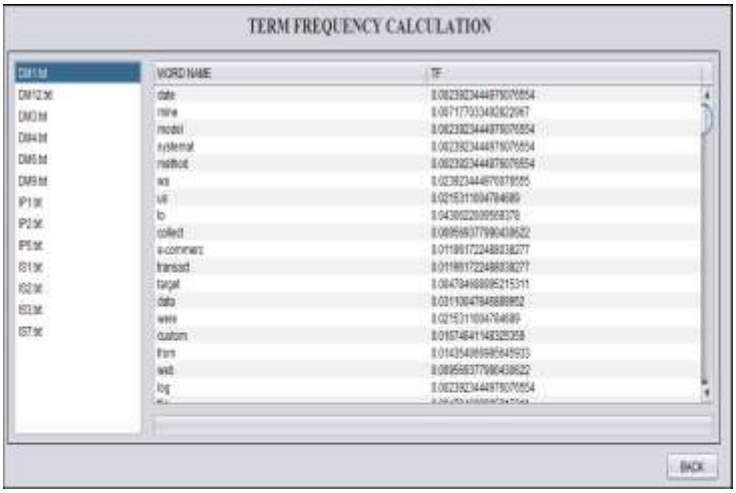
Fig. 1. Flow of Proposed Work'

IV. METHODOLOGY

Design and Implementation: The proposed work is designed to accept the data from user including large sets of text documents, apply the pre-processing on the text file then clustering it using clustering algorithms. The clustering is done with reference of sample document which will be match with all other documents and clusters are formed by analysing the difference between documents and the centroids used in clustering algorithms.

A. PREPROCESSING STEPS:

We need to perform some preprocessing steps before executing the clustering algorithms on text datasets. In particular, stop-words like prepositions, pronouns, articles that are irrelevant to document metadata must be removed. The Snowball stemming algorithm for Portuguese words can be used. The documents are represented in a vector space model [15]. Each document in this model is represented by a vector which contains the frequencies of occurrences of words. The dimensionality reduction technique known as Term Variance (TV) [16] to increase the effectiveness and efficiency of clustering algorithms. The words having great variances over the documents from attributes are selected. To compute distances between documents, two measures are being used, namely: cosine-based distance [15] .The other have been used to calculate distances between file (document) names only.



TERM FREQUENCY CALCULATION

DM1.txt
DM12.txt
DM3.txt
DM4.txt
DM5.txt
DM9.txt
IP1.txt
IP2.txt
IP5.txt
IS1.txt
IS2.txt
IS3.txt
IS7.txt

WORD NAME	TF
date	0.0023923444976076554
mine	0.007177033492822967
model	0.0023923444976076554
systemat	0.0023923444976076554
method	0.0023923444976076554
ws	0.023923444976076555
ui	0.0215311004784689
to	0.0430622009569378
collect	0.009569377990430622
e-commerc	0.011961722488038277
transact	0.011961722488038277
target	0.004784688995215311
data	0.03110047846889952
were	0.0215311004784689
custom	0.01674641148325358
from	0.014354066985645933
web	0.009569377990430622
log	0.0023923444976076554

Fig. 2. Term Frequency calcution example.

B. CLUSTERING ALGORITHMS:

Partitional K-means and Improved K-means [2] are famous algorithms in the machine learning and data mining fields, and therefore they have been used in our study.

K-means Algorithm: K-means starts with selection of K randomly chosen objects as initial clusters centers, named as seeds. The cluster centers are moved around in space in order to minimize the RSS. This two steps are repeated iteratively until a stopping criterion is met.

- Reassignment of objects is done to the cluster with the closest centroids.
- Each centroid is recomputed based on the current members of its cluster.

The termination conditions as stopping criterion are:

- The number of iterations are equal to a pre-decided value for number of iterations to be completed.
- The centroids μi are not toggling between iterations.
- Termination of algorithm when the RSS value falls below a pre-estabilished threshold.

Porter Stemmer Algorithm: The algorithm dates from 1980. Stems using a set of rules, or transformations, applied in a succession of steps. It applies 60 rules in 6 steps. It is default go-to stemmer overview is as follows:

- Step 1: Gets rid of plurals and '-ed' or '-ing' suffixes.
- Step 2: If there is another vowel in the stem then turns the terminal y to i .
- Step 3: Maps double suffixes to single one: '-ization',' –ational', etc.
- Step 4: Deals with suffixes like '-full',' –ness' etc.
- Step 5: Takes off '-ant', '-ence', etc.
- Step 6: Removes a final –e.

Cosine Similarity Measure: Measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.

The cosine of 0° is 1 and for any other angle it is less than 1. It is a judgement of orientation and not of magnitude:

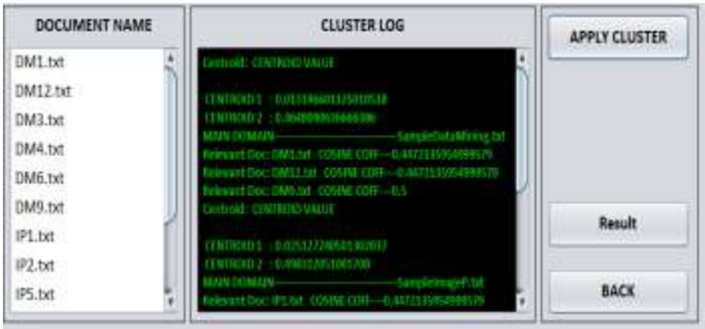
- Hence, the two vectors with the same orientation have a cosine similarity of 1.
- The two vectors at 90° and two vectors diametrically have a similarity of 0.
- Opposed have a similarity of - 1independent of their magnitude.

Cosine similarity is used in positive space particularly, where the outcome is neatly bounded in [0,1].

Improved K-Means: Algorithm 2: The Enhanced Method

Require: D = {d1, d2, d3,..., di,..., dn }(Set of n data Points).The di = { x1, x2, x3,..., xi,..., xm }(Set of attributes of one data point).The k (Number of desired clusters).Ensure: A set of k clusters.

```
Steps:
1: In the given data set D, if the data points contains the both
   positive as well as negative attribute values then go to step 2,
   otherwise go to step 4.
2: To find the minimum attribute value in the given data set D.
3: Subtract with the minimumattribute value for each data point attribute .
4: Calculating the distance from origin for each data point.
5: Sort the distances obtained in the step 4 and also sort the data points in accordance with the distances.
6: The sorted data points are partitioned into k equal sets.
7: The middle point is taken as the initial centroid from each set.
8: The distance between each data point is computed as di (1 <= i<= n) to all the initial centroids cj (1 <= j <= k).
9: Repeat
10: Find the closest centroid cj for each data point di and assign di to cluster j.
11: Set ClusterId[i]=j. // j:Id of the closest cluster.
12: Set NearestDist[i]= d(di, cj).
13: For each cluster j (1 <= j <= k), recalculate the centroids.
14: For each data point di,
14.1 Compute its distance from the centroid of the present
   nearest cluster.
14.2 If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster.
     Else
14.2.1 For every centroid cj (1<=j<=k) compute the
     distance d(di, cj).
       End for;
Until the convergence criteria is met.
```



```
Centroid: CENTROID VALUE

 CENTROID 1  : 0.0131946601175010518
 CENTROID 2  : 0.0648090631666306
MAIN DOMAIN-----------------------SampleDataMining.txt
Relevant Doc: DM1.txt  COSINE COFF---0.4472135954999579
Relevant Doc: DM12.txt  COSINE COFF---0.4472135954999570
Relevant Doc: DM9.txt  COSINE COFF---0.5
Centroid: CENTROID VALUE

 CENTROID 1  : 0.0251272[illegible]
 CENTROID 2  : 0.09801[illegible]
MAIN DOMAIN-----------------------SampleImageIP.txt
Relevant Doc: IP1.txt  COSINE COFF---0.4472135954999579
```

Fig. 3. EXAMPLE OF DOCUMENT CLUSTERING

V. EXPERIMENTAL EVALUATION

We presented an approach that applies document clustering methods for analysis of text documents. Also, we reported and discussed several practical results that can be very useful for researchers and practitioners. In our experiments the partitional algorithms known as K-means and Improved K-means presented the best results. They provide summarized views of the documents being inspected and are helpful tools for examiners that analyze textual documents from seized computers or any other organizations. Good results are obtained if proper initialization is done to K-means. In addition, some of our results suggest that using the file names along with the document content information may be useful for cluster ensemble algorithms. Most importantly, we observed that clustering algorithms indeed tend to induce clusters formed by either relevant or irrelevant documents to enhance the expert examiner's job of analysis. Our evaluation of the proposed approach in some real-world applications shows that it has the potential to speed up the computer inspection process.

The labels assigned to clusters may enable the expert examiner to identify the semantic content of each cluster more quickly eventually even before examining their contents.

A. ADVANTAGES

Most importantly, the clustering algorithms indeed tend to induce clusters formed by either related or unrelated documents, thus contributing to improve the domain examiner's job. Furthermore, proposed approach in applications show that it has the ability to speed up the computer inspection process. Its main application is to reduce efforts of reading each and every document in detail. Since, the labels or information of previous datasets cannot be used each time the new dataset is used with new types of classes. Hence, there is a need of dynamic clustering which can be done with the proposed system. Any text document can be clustered.

B. LIMITATIONS

Success of any clustering algorithm is data independent so scalability may be an issue. Dataset must be too large to be clustered. The format of document should be of text type only.

C. APPLICATIONS

The application of our proposed system can be in data classification and reference matching applications. We use the proposed system to input large set of documents in text format from any source and to cluster them according to our requirement using sample data file for reference. We can cluster domain related information for analysis. This proposed system can be useful in any kind of text data analysis domains.

VI. FUTURE SCOPE

With using different types of algorithms we can check for accuracy of product. For example, by using cosine-based distance and Leven-shtein-based distance algorithms we are computing distances between documents. Application of document clustering can be categorized to two types online and offline.

VII. CONCLUSION

We can use document clustering on a large dataset of research papers as input to our project and reduce the efforts of reading each and every document for analysis which would be beneficial for an organization working in relevance of research papers.

Using this proposed approach which can become an ideal application for document clustering to research paper analysis. There are several practical results based on our work which are extremely useful for the experts working in sorting documentation department.

We presented an approach that applies document clustering methods to forensic analysis of computers. This approach can be very useful for researchers and practitioners of organization relevant to working with text documents.

VIII. ACKNOWLEDGMENT

We would like to offer our sincere thanks to our guide Prof. Priya Thakare for their valuable guidance.

We would also like to thank Cybernetics Software Pvt. Ltd. for their sponsorship and guidance.

IX. REFERENCES

[1]. Luis Filipe da Cruz Nassif and Eduardo RaulHruschka, "Document clustering for forensic computing: An approach for improving computer inspection," -IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 1, JANUARY 2013.

[2]. "Document Clustering For Computer Inspection,"-International Journal of Engineering and Technical Research (IJETR) ISSN: 2321 - 0869, Volume – 3 , Issue - 1, January 2015.

[3]. L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on unsupervised feature selection methods for text clustering," in Proc. IEEE Int. Conf. Natural Language Processing and Knowledge Engineering, 2005, pp. 597–601.

[4] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London, U.K.: Arnold, 2001.

[5] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Engle- wood Cliffs, NJ: Prentice-Hall, 1988.

[5] L. Kaufman and P. Rousseeuw, *Finding Groups in Gata: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley-Interscience, 1990.

[6] R. Xu and D. C. Wunsch, II, *Clustering*. Hoboken, NJ: Wiley/IEEE Press, 2009.

[7] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse frame- work for combining multiple partitions," *J. Mach. Learning Res.*, vol.

3, pp. 583–617, 2002.

[8] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro, “Evolving clusters in gene-expression data,” *Inf. Sci.*, vol. 176, pp. 1898–1927,
2006.

[9] B. K. L. Fei, J. H. P. Eloff, H. S. Venter, and M. S. Oliver, “Exploring forensic data with self-organizing maps,” in *Proc. IFIP Int. Conf. Dig- ital Forensics*, 2005, pp. 113–123.

[10] N. L. Beebe and J. G. Clark, “Digital forensic text string searching: Im- proving information retrieval effectiveness by thematically clustering search results,” *Digital Investigation, Elsevier*, vol. 4, no. 1, pp. 49–54,
2007.

[11] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, “Towards an integrated e-mail forensic analysis frame- work,” *Digital Investigation, Elsevier*, vol. 5, no. 3–4, pp. 124–137,
2009.

[12] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, “Mining writeprints from anonymous e-mails for forensic investigation,” *Dig- ital Investigation, Elsevier*, vol. 7, no. 1–2, pp. 56–64, 2010.

[13] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, “Text clustering for digital forensics analysis,” *Computat. Intell. Security Inf. Syst.*, vol. 63, pp. 29–36, 2009.

[14] K. Stoffel, P. Cotofrei, and D. Han, “Fuzzy methods for forensic data analysis,” in *Proc. IEEE Int. Conf. Soft Computing and Pattern Recog- nition*, 2010, pp. 23–28.

[15] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, “Relative clustering validity criteria: A comparative overview,” *Statist. Anal. Data Mining*, vol. 3, pp. 209–235, 2010.

[16] L. Liu, J. Kang, J. Yu, and Z. Wang, “A comparative study on unsu- pervised feature selection methods for text clustering,” in Proc. IEEE Int. Conf. Natural Language Processing and Knowledge Engineering,
2005, pp. 597–601.

[17] V. Levenshtein, “Binary codes capable of correcting deletions, inser- tions, and reversals,” Soviet Physics Doklady, vol. 10, pp. 707–710,
1966.

[18] B. Mirkin, Clustering for Data Mining: A Data Recovery Approach.
London, U.K.: Chapman & Hall, 2005.

[19] A. L. N. Fred and A. K. Jain, “Combining multiple clusterings using evidence accumulation,” IEEE Trans. Pattern Anal. Mach. Intell., vol.
27, no. 6, pp. 835–850, Jun. 2005.

[20] L. Hubert and P. Arabie, “Comparing partitions,” J. Classification, vol.
2, pp. 193–218, 1985.

[21] C. M. Bishop, Pattern Recognition and Machine Learning. New
York: Springer-Verlag, 2006.

[22] S. Haykin, Neural Networks: A Comprehensive Foundation. Engle- wood Cliffs, NJ: Prentice-Hall, 1998.

[23] L. F. Nassif and E. R. Hruschka, “Document clustering for forensic computing: An approach for improving computer inspection,” in Proc. Tenth Int. Conf. Machine Learning and Applications (ICMLA), 2011, vol. 1, pp. 265–268, IEEE Press.

[24] , Aggarwal, C. C. Charu, and C. X. Zhai, Eds., “Chapter 4: A Survey of Text Clustering Algorithms,” in Mining Text Data. New York: Springer, 2012.

[25] Y. Zhao, G. Karypis, and U. M. Fayyad, “Hierarchical clustering algo- rithms for document datasets,” Data Min. Knowl. Discov., vol. 10, no.
2, pp. 141–168, 2005.

[26] Y. Zhao and G. Karypis, “Evaluation of hierarchical clustering algo- rithms for document datasets,” in Proc. CIKM, 2002, pp. 515–524.

[27] S. Nassar, J. Sander, and C. Cheng, “Incremental and effective data summarization for dynamic hierarchical clustering,” in Proc. 2004
ACM SIGMOD Int. Conf. Management of Data (SIGMOD ’04), 2004, pp. 467–478.

[28] K. Kishida, “High-speed rough clustering for very large document col- lections,” J. Amer. Soc. Inf. Sci., vol. 61, pp. 1092–1104, 2010, doi:
10.1002/asi.2131.

[29] Y. Loewenstein, E. Portugaly, M. Fromer, and M. Linial, “Effcient al- gorithms for exact hierarchical clustering of huge datasets: Tackling the entire protein space,” Bioinformatics, vol. 24, no. 13, pp. i41–i49,
2008.