International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-1, January 2015

Document Clustering For Computer Inspection

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

Abstract— Clustering is an approach that organizes a large quantity of unstructured text documents into a small number of meaningful and coherent clusters. We compare and analyze the effectiveness of these measures in partitional clustering for text document datasets. Clustering means extraction and fast information retrieval or filtering. Related to document clustering, clustering methods can be used to automatically group the retrieved documents into a list of useful categories. Document clustering contains descriptors and descriptor retrival. Descriptors are collection of words that describe the contents of the cluster. Document cluster is generally considered to be a centralized process.

***Index Terms*— 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 analysis and interpretation of data. Therefore, methods for automated data analysis, 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 little knowledge about data is present [2] [3]. It focuses on clustering the text documents documents using various 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. PROPOSED SYSTEM OBJECTIVES

The project aim is to cluster the documents of research papers to assure some organization for the originality or relativity of each document with their required domain of interest and reduce the effort of inspecting each and every document. We use recent clustering algorithms or techniques to cluster documents and find the number of clusters too.

Features obtained by proposed system are:

Correctness

Application should be correct in terms of its calculations, functionality used internally and the navigation should be correct. This means application should adequate for functional requirements.

Maintainability

Different versions of the product should be maintained easily. Development should be easy to add code to existing system, should be upgraded easily for new features and new technologies time to time. Maintenance should be affordable and easy. System should be easy to maintain and correcting defects or making a change in the software.

Reliability

Measure if product is reliable enough to sustain in any condition. System should give correct results. Reliability of product is measured in terms of working of the proposed system under different working environment and different conditions.

Robustness

It is the ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc.

Usability

This can be measured in terms easy usability. Application should be user friendly. Project should be learn easily. Navigation should be simple.

III. BACKGROUND

Clustering algorithms have been studied for many years and the literature on the subject is huge. Therefore, we decided to select a set of (six) representative algorithm in order to show the potential of the proposed approach, are *partitional K-means and K-medoids,* the hierarchical Single/Complete/Average Link, and the cluster ensemble algorithm known as CSPA(Clustering based Similarity Partition Algorithm). These algorithms were executed with different combinations of parameters, results in sixteen different algorithm instantiations. 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.

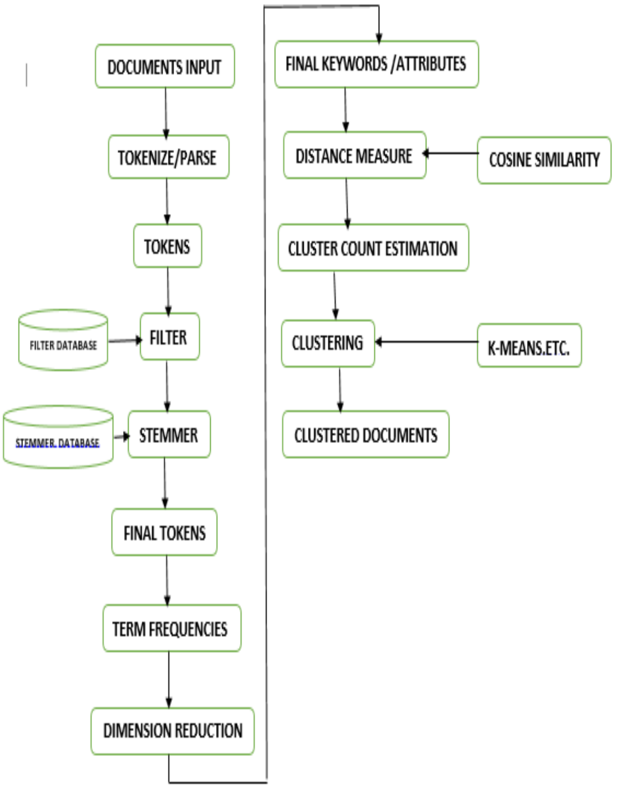
To analyzing research paper we are providing administrative class. Admin can have access to all functions such as uploading the documents, all pre-processing steps, display the scores of documents, clustering and at the last showing result. Pre-processing steps involved parsing, filtering, stemming, 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.

Manuscript received January 22, 2015.

Prof. Priya Thakare, Rekha Kamble, Priyanka Karche, Sneha Gaikwad, Manish Khaladkar, Sinhgad Institute of Technology and Science, Narhe, Pune 4,University of Pune, INDIA

IV. FLOW OF APPLICATION



V. CLUSTERING ALGORITHMS AND PREPROCESSING

A. PREPROCESSING STEPS:

We need to perform some preprocessing steps before executing the clusteing algorithms on text datsets. In particular, stopwords like prepositions, pronouns, articles, and irrelevant document metadata must be removed. Tthe 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] and Levenshtein-based distance [17]. The other have been used to calculate distances between file (document) names only.

B. CLUSTERING ALGORITHMS:

The clustering algorithms used in our study the partitional K-means [2] and K-medoids [4], the hierarchical Single/Complete/Average Link [5], and the cluster ensemble based algorithm known as CSPA [6] are famous in the ma- chine learning and data mining fields, and therefore they have been used in our study.Our choices regarding their use in our proposed system deserve further comments.For instance, the K-means and K-medoids are similar to each other. However, instead of computing centroids, it uses medoids, which are the representa- tive objects of the clusters. This property makes it particularly interesting for applications in which (i) centroids cannot be computed; and (ii) distances between pairs of objects are available, as for computing dissimilarities between names of documents with the Levenshtein distance [17]. Considering the partitional algorithms, it is widely known that both K-means and K-medoids are sensitive to initialization and usually converge to solutions that represent local minima. Trying to minimize these problems, we used a nonrandom initialization in which distant objects from each other are chosen as starting prototypes [18]. Unlike the partitional algorithms such as K-means/medoids, hierarchical algorithms such as Single/ Complete/Average Link provide a hierarchical set of nested par- titions [3], usually represented in the form of a dendrogram, from which the best number of clusters can be estimated. In par- ticular, one can assess the quality of every partition represented by the dendrogram, subsequently choosing the one that provides the best results [14]The CSPA algorithm [6] essentially finds a consensus clustering from a cluster ensemble formed by a set of different data partitions. More precisely, after applying clustering algorithms to the data, a similarity (coassociation) matrix [19] is computed. Each element of this matrix represents pair-wise similarities be- tween objects. The similarity between two objects is simply the fraction of the clustering solutions in which those two objects lie in the same cluster. Later, this similarity measure is used by a clustering algorithm that can process a proximity matrix—e.g., K-medoids—to produce the final consensus clustering. The sets of data partitions (clusterings) were generated in two different ways: (a) by running K-means 100 times with different sub- sets of attributes (in this case CSPA processes 100 data parti- tions); and (b) by using only two data partitions, namely: one obtained by K-medoids from the dissimilarities between the file names, and another partition achieved with K-means from the vector space model. In this case, each partition can have dif- ferent weights, which have been varied between 0 and 1 (in in- crements of 0.1 and keeping their sum equals to1).it repeatedly for an increasing number of clusters. For each value of , a number of partitions achieved from different initializations are assessed in order to choose the best value of and its corresponding data partition, using the Silhouette [4] and its simplified version [7], which showed good results in [14] and is more computationally efficient. In our experiments, we assessed all possible values of in the interval where is the number of objects to be clustered.

VI. ADVANTAGES& LIMITATIONS

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.Unlike other clustering systems we can calculate number of clusters in our system.

B. LIMITATIONS

Success of any clustering algorithm is data independent so scalability may be an issue. Bisecting k-means algorithms can also induces dendrograms. Dataset must be too large to be clustered. The format of document should be of text type only.

VII. APPLICATION

The applications of our proposed system can be in data classification and reference matching applications.

VIII. 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.

IX. CONCLUSION

Hence, 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. More specifically, in our experiments the hierarchical algorithms known as Average Link and Complete Link presented the best results.

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.

REFERENCES

[1] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The expanding digital universe: A forecast of worldwide information growth through 2010," *Inf. Data*, vol. 1, pp. 1–21, 2007.

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

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

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

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

[6] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," *J. Mach. Learning Res.*, vol. 3, pp. 583–617, 2002.

[7] 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.

[8] 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. Digital Forensics*, 2005, pp. 113–123.

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

[10] 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.

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

[12] 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.

[13] K. Stoffel, P. Cotofrei, and D. Han, "Fuzzy methods for forensic data analysis," in *Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition*, 2010, pp. 23–28.

[14] 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.

[15] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[16] 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.

[17] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, 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. Englewood 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 algorithms 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 algorithms 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 collections," 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 algorithms for exact hierarchical clustering of huge datasets: Tackling the entire protein space," Bioinformatics, vol. 24, no. 13, pp. i41–i49, 2008.

Luís Filipe da Cruz Nassif received the B.Sc. degree in computer engineering from the Military Institute of Engineering (IME), Brazil, in 2005, and the M.Sc. degree in electrical engineering from the University of Brasilia (UnB), Brazil, in 2011.
Since 2006, he has been working as a computer forensic examiner with the Brazilian Federal Police Department, São Paulo, Brazil. His main research in- terests are in computer forensics and data mining.

Eduardo Raul Hruschka received the Ph.D. degree in computational systems from COPPE/Federal Uni- versity, Rio de Janeiro, Brazil, in 2001. He is with the Computer Science department (ICMC), University of São Paulo (USP) at Carlos, Brazil. From 2010 to 2012, he was a visiting researcher at the University of Texas at Austin. His primary research interests are in data mining and machine learning. He has authored or coauthored more than 50 research publications in peer-reviewed reputed journals and conference proceedings.

Dr. Hruschka is associate editor of *Information Sciences* (Elsevier). He has also been a reviewer for several journals such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNTICS,IEEE TRANSACTION ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, as well as a member of the Program.