Volume 4, Issue 10, October 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Conceptual Document Clustering Using Sequential Patterns

Ms. Ajita Sawant*
Computer Dept. & Pune University
Pune, India

Prof. R. N. Phursule
Computer Department& Pune University
Pune, India

Abstract— Document clustering has shown to be very useful for computer inspection. Number of files are examined in computer analysis. That data of files consists unstructured text. The majority of tools available on the market have the ability to permit investigators to analyze data that was gathered from a computer system. For computer examiners It's quite difficult. Here the great interest is in automated methods of analysis. In particular, clustering algorithms for documents can facilitate the knowledge which discovered new and useful from the documents under analysis. The largest collection of data of different file format is called World Wide Web .The principal media of business and academic information belonging to Electronic documents. For the clustering of documents different types of clustering algorithms are used. The aim of making the decisions, making process performed by examiners more efficient Self-Organizing Maps (SOM) based algorithms are used for clustering files. In related application domain, e-mails are grouped by using lexical, syntactic, structural and domain specific features.

Since most existing text mining methods all suffer from the problems of polysemy and synonymy because they adopted term-based approaches. From this statement, we can conclude that since the term-based approach, the clustering might also suffer from achieving high conceptualization. We also can state that a phrase-based or sequential pattern based clustering can be effective for finding conceptually relevant cluster than the term-based clustering because we get a good conceptualization within a phrase or within an ordered set of terms than a single term. Here our work is inspired by motivation of finding conceptual clusters among the given document set.

Keywords— Document Clustering; Sequential Patterns; K-Means Algorithm; Single Link Algorithm

I. INTRODUCTION

Clustering algorithms are typically used for exploratory data analysis, where there is little or no prior knowledge about the data. This is precisely the case in several applications of Computer Forensics, including the one addressed in our work. From a more technical view point, our datasets consist of unlabeled documents the classes or categories of documents that can be found are a priori unknown. Moreover even assuming that labeled datasets could be available from previous analysis there is almost no hope that the same classes (possibly learned earlier by a classier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computer sand associated to different investigation processes. More precisely, it is likely that the new data sample would come from a different population. In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner.[1]

The rationale behind clustering algorithms is that documents within a valid cluster are more similar to each other than they are to documents belonging to a different cluster.[3] Thus, once a data partition has been induced from da-ta, the expert examiner might initially focus on reviewing one may eventually decide to scrutinize other documents from each cluster. By doing so, one can avoid the hard task of examining all the documents (individually) but,even if so desired, it still could be done. Though some of the previous clustering algorithms are better at their perspective, these algorithms suffer from synonymy and polysemy. These algorithms found to be based on keyword similarity matching. Frequencies or weights of the keywords are used as input to those, so they may not good enough to find conceptually similar documents. Here we propose a document clustering algorithm to find conceptual cluster by using sequential patterns. One can easily conclude that instead of using single keyword based approach for finding similarity, if an ordered sequence of keyword will be more effective to find conceptual match among text documents.

II. LITERATURE SURVEY

Non-hierarchical methods The non-hierarchical meth-ods are heuristic in nature, since a priori decisions about the number of clusters, cluster size, criterion for cluster member-ship, and form of cluster representation are required. Since the large number of possible divisions of N documents into M clus-ters makes an optimal solution impossible, the non-hierarchical methods at- tempt to find an approximation, usually by partitioning the data set in some way and then reallocating documents until some criterion is optimized. The computational requirement O(NM) is much lower than for the hierarchical methods if M ¡¡ N, so that large data sets can be partitioned. The non-hierarchical methods were used for most of the early work in document clustering when computational resources were limited; see for example work on the SMART project, described by Salton (1971)[2].

Hierarchical methods Most of the early published work on cluster analysis employed hierarchical methods (Blashfield and Aldenderfer 1978), though this was not so in the IR field. With im- provements in computer resources, the easy availability of software packages for cluster analysis, and improved algorithms, the last decade of work on clus- tering in IR retrieval has concentrated on the hierarchical agglomerative clustering methods (HACM, Willett [1988]).

III. MOTIVATION

After a careful analysis of the literature as mentioned in the earlier section, we came to the conclusion that this particular domain still has potential for improvement. The accurate pattern mining will help the retrieval of conceptually se-mantic information from a large. Apart from offering the imme-diate and tangible benefits like reduced information retrieval for fired query it is beneficial to the customer to retrieve con-ceptual information only.

A. Problem Statement

To design and implement Efficient Pattern selection algo-rithm that is used in multi- dimensional data using cluster- ing for improving the quality as well as performance of Pattern selection algorithm.

B. Problem Scope

Develop a system for

1. Removing Irrelevant data in data set.
2. Removing Redundant data in data set.
3. Improve the performance of algorithm based on execution time.
4. To improve clustering based on Pattern matching

IV. IMPLEMENTATION DETAILS

A. Pre-Processing

Steps: a. Stop word Removal
b. Non Word Removal
c. Stemming

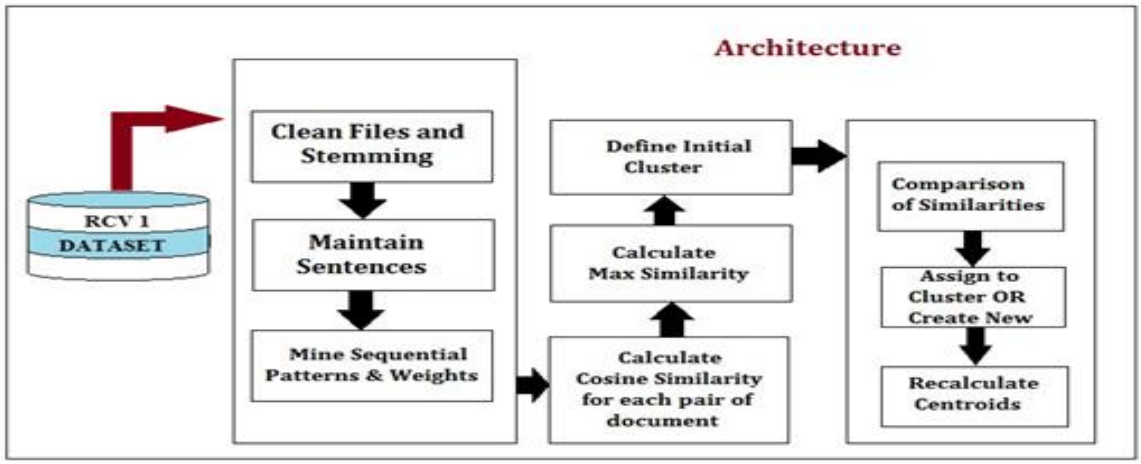
Here in our proposed algorithm we require a pre-processed dataset with maintaining the sentences as it is.

B. Final Finding Sequential Patterns:

Here we propose a simple and effective technique to find Sequential Pattern within a document.

Let we divide a file in sentences. Now for each term within each sentence we can calculate the co-occurrence of a term with its very next term. This can be done within a couple of nested 'for' loops. Here in our proposed algorithm, we limit our patterns' length up to 3 to avoid excessive looping. And we can analyze that patterns' of length 3 are much and more sufficient for conceptualization.

C. Block Diagram:



D. Inputs:

1) Text document dataset
2) Sequential pattern and weight pairs

Process:

Sim (Di, Dj) := Di ∩ Dj / Di∪ Dj
Simmax:= Max{ Simij }
Cluster :=0
Ccluster:= {Di, Dj}
…. where Simij(Di,Dj) = Simmax

Ccluster->centroid :=Simmax
Simmean:= ∑ (CosineSim(Dr, Dcluster)) / |Ccluster| …… where |Ccluster| is total documents in Cluster
For all Drϵ (D-{Di, Dj}) do
cluster:= cluster-> centroid +∑simrk/ |simrk|+1
Output:
Clusters input dataset.

E. Proposed Algorithm:

1. For I = 0 to n-1
2. Do
3. For j = i+1 to n-1
4. Do
5. Simij:= CosineSim(Di,Dj)
6. End for
7. End for
8. Simmax:= max{ Simij } // Similarity of Di and Dj which is maximum than any other pair of docu-ment.
9. cluster := 0
10. Ccluster:= {Di, Dj}
11. Ccluster->centroid := Simmax
12. For each Dr // Dr are the remaining document i.e r != i and r != j
13. Do
14. For each Ccluster
15. Do
16. Simmean:= ∑ (CosineSim(Dr, Dcluster, k)) / |Ccluster| // |Ccluster| is total documents in Cluster
17. If Simmean>= Ccluster->centroid
18. Ccluster:= CclusterUDr
19. Ccluster->centroid := recalculate centroid(Ccluster)
20. Else
21. cluster= cluster+1
22. Ccluster:= {Dr}
23. Ccluster->centroid := calculate centroid(Ccluster)
24. End for
25. End for

F. Mathematical model for proposed work:

The mathematical equation to calculate cosine similarity is:

$$Sim(N_i, N_j) = \frac{\sum N_{iN_i} \times N_{iN_j}}{\sqrt{(\sum N_{i_{N_i}})^2} \times \sqrt{(\sum N_{i_{N_j}})^2}}$$

Set Theory:
For finding sequential patterns

1. D is a set of input Documents
   D= {Di,i | 0<i< No. of Documents}
2. T is a set of all the unique terms
   T={Ti,i , 0<i< No. of Documents, j ∈ T}
3. SP is a set of sequential patterns
   SPi,i ={Ti,i ,0<i< No. of Documents, j ∈ T}

For Document Clustering:

1. SP is a set of sequential patterns
   SPi,i ={Ti,i , 0<i<No.of Documents, ∈T}
   Simi,i ={Simi,i , 0<i< No. of Documents, 0<j< No. of Docu-ments,}
2. Sim is a set of similarities among the document
3. Cen is set of Centroids
   Cen={Ceni, I ∈ Cen}
4. C is set of clusters
   C={Ci, i∈ C}

If we consider the proposed system as G, then we can define G in a set theory as
G={D, SP, Sim,Cen,C}
Where D= all the input documents

SP= All the sequential patterns Sim= Similarity between a peer of document
Cen = Centroid of a cluster
C= Set of Cluster

V. RESULT AND ANALYSIS

Accuracy for Results of K-means and Results of proposed method Over Iris Flower Dataset:

TABLE I: ACCURACY

No. of Cluster (K)	3	4	5	6	7	8	9	10
K-means	0.889	0.693	0.666	0.666	0.66	0.66	0.6	0.6
Proposed method	0.890	0.695	0.667	0.668	0.67	0.67	0.6	0.6

TABLE III: EXECUTION TIME

Trial No	K-Means (SEC)	Proposed Method (SEC)
1	0.362	0.361
2	0.011	0.010
3	0.007	0.006
4	0.005	0.004
5	0.008	0.007
6	0.006	0.005
7	0.007	0.006
8	0.009	0.008
9	0.023	0.022
10	0.013	0.011

TABLE IIIII: PRECISION

Query No	K-Means (SEC)	Proposed Method (SEC)
1	0.37	0.39
2	0.31	0.32
3	0.51	0.52
4	0.95	0.97
5	0.59	0.60
6	0.61	0.62
7	1.0	1.01
8	0.48	0.50
Average Precision	0.60	0.61

Time Complexity Analysis:

Time complexity of Single Link= O(n2)

Time complexity of K-means= O(n × K × I × d) where I is no. of Iterations taken.

In our proposed algorithm Time complexity is O(n2) As for calculating Similarity for each pair will take 2 loops and assigning clusters will take 2 loops so at the worst case it will take O(n2) time. As in K-means effect of stopping crite-ria or iteration degrades the performance with respect to time.

We have time complexity similar to Single Link, but still we have the clusters with high conceptual meaning.

Efficiency:

We have time complexity similar to Single Link, but still we have the clusters with high conceptual meaning.

Redundancy:

In K-means, the algorithm keeps executing after forming the actual clusters (Executes until meeting of stopping criteria), so it has redundancy in the form of loop executions or calcu-lating same cluster until stopping criteria. In our proposed algorithm we do not keep execution until stopping criteria, because we did not have any stopping criteria. Execution of loops gives the final clusters. So we avoid the execution re-dundancy.

Availability:

In K-means or K-medoids, we give the no. of cluster as input, so there is a chance of NULL clusters. For example we have given no. of clusters=5 and we have only 7 documents to cluster, and then there is chance of finding Null clusters. Here in our proposed algorithm the no. of clusters are calcu-lated by algorithm itself. So there no chance of finding NULL clusters. So we have a good availability on terms of clusters.

Reliability:

As the K-means algorithm is NP-Hard (we cannot guess when the stopping criteria will meet), so its reliability is not so good. Our proposed algorithm we did not have any stop-ping criteria so its a NP-complete algorithm, so we have a good reliability than K-means.

VII. CONCLUSIONS

As previous clustering algorithms suffer from polysemy and synonymy, the proposed algorithm is able to cluster the documents which are conceptually similar. Here in our pro-posed algorithm semantics are preserved as we use sequential orordered term set. Semantics can be further maintained by increasing length of term set. The proposed algorithm is capable to cluster the input documents in more conceptual clusters..

VII. FUTURE ENHANCEMENT

The more effective technique such machine learning algorithm for finding sequential patterns can be adopted. Dictionary of found sequential patterns may be created for Sub-sequent processing to achieve more time efficiency. As the number of clusters is not user defined, the number of cluster may increase as sparse document increase, so we can move towards predefined number of cluster method.

REFERENCES

[1] Filipe daCruz, Nassif and Eduardo Raul Hruschka, "Document Clustering for Forensic Analysis: An Approach for Improving Computer Inspection"IEEE,VOL 8.No. 1, January 2013.

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

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

[4] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood 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 framework for combining multiple partitions," *J. Mach. Learning Res.*,vol. 3, pp. 583–617, 2002.

