International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 4, Issue 1 (January 2015), PP.08-12

Discovering Novel Information with sentence Level clustering From Multi-documents

Mr. Uday. D. Patil[1], Mrs. Sharmila M. Shinde[2]
[1]Computer Engineering Department, JSCOE, Pune University, India
[2]Head of Computer Engineering Department, JSCOE. Pune University, India

Abstract:- Specific objective to discover some novel information from a set of documents initially retrieved in response to some query. Clustering sentences level text, effective use and update is still an open research issue, especially in domain of text mining. Since most existing system uses pattern belong to a single cluster. But here we can use patterns belongs to all cluster with different degree of membership. Since sentences of those documents we would expect at least one of the clusters to be closely related to the concepts described by the query term. This paper presents a Novel Fuzzy Clustering Algorithm that operates on relational input data (i.e. data in the form of square matrix of pair wise similarities between data objects).

Keywords:- Fuzzy relational clustering, natural language processing, graph centrality.

I. INTRODUCTION

Knowledge discovery and data mining have attracted a great deal of attention and we need to utilize such data into useful information and knowledge such information and knowledge can be applied at market analysis and business management it will lead to great benefit. Basically knowledge discovery is nothing but process of extraction of information from large documents. It is a challenging issue to find appropriate and accurate knowledge in a text document to help to user what they want. In the beginning, Information Retrieval, provided many term based methods to solve challenge, such as Roccio and Probabilistic models, rough sets models, BM25 and support vectors machine (SVM), based filtering models. From past few year, there are several data mining techniques have been presented in order to perform different knowledge tasks. It includes association rule mining, frequent itemset mining, sequential pattern mining, maximum pattern mining, closed pattern mining. Sentence may contain more than one topics or issue present within documents or a set of documents. However because of most sentence similarities measures do not represent sentences in a common metric. Traditional approach based on Prototype or mixture of Gaussian is not suitable for sentence clustering. While clustering text at the document level where documents are represented as data point in a high dimensional vector space in which each row represent documents and column represent attributes of those documents. Since pair wise similarity and dissimilarity can be easily identified from the attribute data using similarity measures such as the word co-occurrence. However similarity can be calculated in term of word co-occurrence and it is valid at the document level only and it will not check small-sized text fragments such as sentences. Since two or more sentences may be semantically related having few or if any word in common. Semantic information treated in external source such as word Net. Interestingly, the notion of fuzzy partitioning based on relational data is not new, and can be traced to the late nineteen sixties approximately the same time as which the prototype-based k-Means and Iso data algorithms were first Introduced. Ruspini proposed an optimization scheme based on iteratively minimizing an objective function based On pair wise dissimilarity data.

II. RELATED WORK

Most existing text mining methods uses term based approach and pattern matching methods. From many years pattern mining has been extensively studied in data mining community. A variety of algorithm proposed such as Apriori Algorithm, prefix span, FP-tree, SPADE, and GST, have been proposed. These algorithms mainly concentrate on developing efficient mining algorithms for discovering pattern from a document sets. For the challenging issue closed sequential pattern have been used for the text mining in which proposed that concept of closed pattern in text mining was useful and had the potential for improving the performance of text mining. Naturally language processing (NLP) is a new computational tool is used to identify the meaning of text documents. Recently new Concept–Based Model was invented to overcome the gap between NLP and text mining which analyzed the sentence and documents level. This model includes three part: First component analyzed the semantic structure of the sentence, second component constructed a conceptual ontology graph [COG] to describe semantically structure.

2.1. Pattern Taxonomy Model

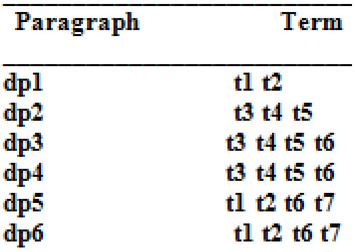
In this paper, we assume that all documents are split into paragraphs. So a given document d yields a set of paragraphs PS(d). Let D be a training set of documents, which consists of a set of positive documents, D+; and a set of negative documents, D-. Let T = {t1, t2, t3……Tm} be a set of terms (or keywords) which can be extracted from the set of positive documents, D+.

2.2. **Frequent and Closed Patterns:**

Given a termset X in document ⌈X⌉is used to denote the Covering set of X for d, which includes all paragraphs i.e. {dp|dp € PS (d), x $\subseteq$dp}
Its absolute support is the number of occurrences of X in Its absolute support is the number of occurrences of X in PS(d) that is SUP_a(X)=| ⌈X⌉|. Its relative support is the fraction of the paragraphs that contain the pattern, that is $Sup_{(X)}$ = (| ⌈X⌉| \ |PS (d)|). x

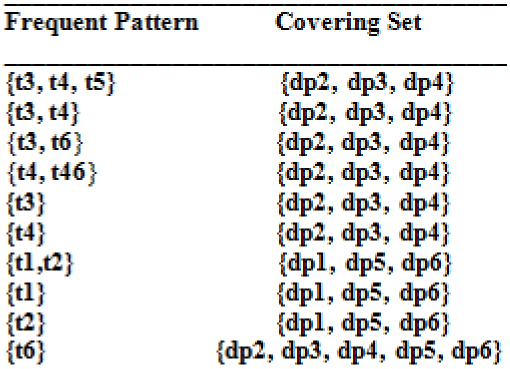
A term set X is called frequent pattern if its sup_r (or sup_a) ≥ min sup, a minimum support.

Table I List a set of paragraph for a given document d



Paragraph	Term
dp1	t1 t2
dp2	t3 t4 t5
dp3	t3 t4 t5 t6
dp4	t3 t4 t5 t6
dp5	t1 t2 t6 t7
dp6	t1 t2 t6 t7

TABLE II Frequent Pattern and Covering Sets



Frequent Pattern	Covering Set
{t3, t4, t5}	{dp2, dp3, dp4}
{t3, t4}	{dp2, dp3, dp4}
{t3, t6}	{dp2, dp3, dp4}
{t4, t46}	{dp2, dp3, dp4}
{t3}	{dp2, dp3, dp4}
{t4}	{dp2, dp3, dp4}
{t1,t2}	{dp1, dp5, dp6}
{t1}	{dp1, dp5, dp6}
{t2}	{dp1, dp5, dp6}
{t6}	{dp2, dp3, dp4, dp5, dp6}

Removed. Let min _sup = 50%, we can obtain ten frequent patterns in Table 1 using the above definitions. Table 2 illustrates the ten frequent patterns and their covering sets. Not all frequent patterns in Table 2 are useful [4]. For Example, pattern {t3,t4}always occurs with term t6 in paragraphs, i.e., the shorter pattern, {t3,t4}, is always a part of the larger pattern, ft3; t4; t6, in all of the paragraphs Hence, we believe that the shorter one, {t3, t4} Hence, we believe that the shorter one, {t3, t4,t6} only. Given a termset X, its covering set ⌈X⌉ is a subset of paragraphs. Similarly, given a set of paragraphs Y $\subseteq$ PS (d), we can define its termset, which satisfies

Termset(Y) = {t|∀dp □ Y=>t □ dp

The closure of X is defined as follow:

Cls(X) = termset (⌈X⌉)

A pattern X (also a termset) is called closed if and only if X=Cls(X). Patterns can be structured into a taxonomy by using the is-a (or subset) relation. For the example of Table 1, where we have illustrated a set of paragraphs of a document, and the discovered 10 frequent patterns in Table 2 if assuming min _sup = 50%. There are, however, only three.

To overcome problem with pattern mining and review the described semantic structure, we use Page rank as general graph centrality measures. A fuzzy relational clustering approach is used to produce clusters with sentences, where each of them corresponds to some content. The output of clustering indicates the strength of the association among the data elements.

III. IMPLEMENTATION

Basically Page Rank can be used more generally to determine the importance of an object in a graph. The key idea behind the Page rank algorithm is that the importance of a node within a graph can be determined by taking account global information recursively computed from the entire graph, with related to high scoring nodes contributing more to the score of a node than connections to low-scoring nodes . Page rank allocated to every node in a directed graph a numerical score between 0 and 1 known as its Page rank (PR).

$$PR\ (V_i) = (1\text{-}d) + d * \sum_{j \in In\ (V_i)} (1/\ (out\ (V_j))\ PR\ (V_j)$$

Where In (Vi) is the set of vertices that point to the Vi, Out (Vj) is the set of vertices pointed to by Vj and d is the damping factor typically set around 0.8 to0.9. The page rank algorithm can easily be modified to deal with weighted undirected edges,

$$PR\ (V_i) = (1\text{-}d) + d * \sum\nolimits^{N}_{J=1} (W_{ij}\ (PR\ (V_j)/ \sum\nolimits^{N}_{K=1} W_{jk}))$$

Where w_{ji} is the similarity between V_j and V_i, and we assume that these weights are stored in a matrix W = $\{W_{ij}\}$ which we refer to as the “affinity matrix.” Note that the summations are now over all objects in the graph.

3.1. Fuzzy Relational Eigenvector Centrality –Based Clustering Algorithm [FRECCA]**:** Instead of traditional Gaussian mixture models, which and covariance’s of mixture component the proposed algorithm uses the Page Rank score of an object within a cluster as a measures of its centrality to that cluster. These Page Rank values are then treated as likelihoods. Since there is no parameterized likelihood function as such, the only parameters that need to be determined are the cluster membership values and mixing coefficients. The algorithm uses Expectation Maximization to optimize these parameters. We assume in the following that the similarities between objects are stored in a Similarity matrix S = $\{S_{ij}\}$, where s_{ij} is the similarity between objects i and j.

Initialization: We assume here that cluster membership values are initialized randomly, and normalized such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialized such that priors for all clusters are equal.

Expectation step: The E-step calculates the PageRank value for each object in each cluster. PageRank values for each cluster are calculated as described in, with the affinity matrix weights W_{ij} obtained by scaling the similarities by their cluster membership values; i.e.

$W_{ij}^{m}=S_{ij}*P_i^{m}*P_j^{m}$

W_{ij}^{m} is the weight between objects i and j in cluster m, sij is the similarity between objects i and j, and p_i^m and $p^m{}_j$ are the respective membership values of objects i and j to cluster m. The intuition behind this scaling is that an object’s entitlement to contribute to the centrality score of some other object depends not only on its similarity to that other object, but also on its degree of membership to the cluster. Likewise, an object’s entitlement to receive a contribution depends on its membership to the cluster. Once PageRank scores have been determined, these are treated as likelihoods and used to calculate cluster membership values. Maximization step. Since there is no parameterized likelihood function, the maximization step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step. The pseudo code is presented in Algorithm1, where $W_{ij,}{}^{m}$ Sij, ${}^{P}{}_{i}{}^{m*}\,{}^{P}{}_{j}{}^{m}$ are defined as above, is the mixing coefficient for cluster m. $PR_i^{m□}$ is the PageRank score of object i in cluster m, and l_i^m is the likelihood of object i in cluster m. $□_{m.}$

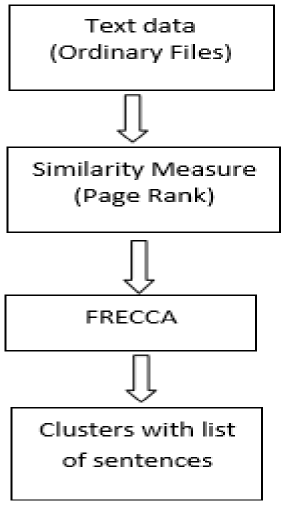


Fig 1: FRECCA Clustering Process

IV. Performance Evaluation

The performance evaluation of the proposed FRECCA clustering algorithm is based on certain performance metrics. The performance metrics used in this paper are Partition Entropy Coefficient (PE), Purity and Entropy, V- Measure, Rand Index and F-Measure. The sentence similarity measure is based on the following metrics.

Purity:
The fraction of the cluster size that the largest class of objects assigned to that cluster.

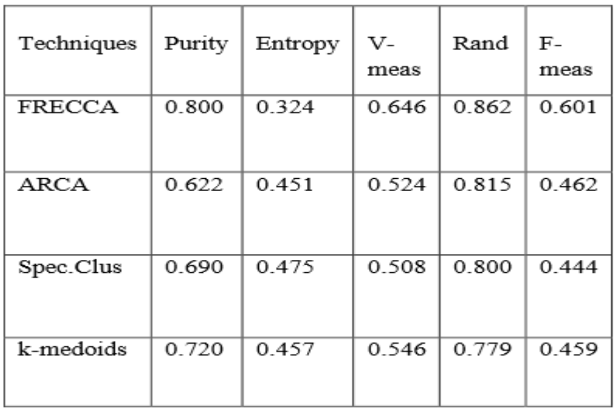
Entropy:
It is a measure of how mixed the objects within the cluster.

***V-measure*:**
It is defined as the harmonic mean of homogeneity and completeness.

Rand Index and F-measure:
It based on a combinatorial approach.

Below in table 1, the comparison is performed out for 6 numbers of clusters. We compare the performance of FRECCA algorithm with ARCA, Spectral Clustering, and k-Medoids algorithms to the quotations data set and evaluating using the external measures. In each algorithm, the affinity matrix was used and pairwise similarities also calculated for each of the method. It is to be observed that FRECCA algorithm is able to identify and avoid overlapping clusters.

Table 3: Clustering Performance Evaluation



Techniques	Purity	Entropy	V-meas	Rand	F-meas
FRECCA	0.800	0.324	0.646	0.862	0.601
ARCA	0.622	0.451	0.524	0.815	0.462
Spec.Clus	0.690	0.475	0.508	0.800	0.444
k-medoids	0.720	0.457	0.546	0.779	0.459

V. CONCLUSION AND FUTURE SCOPE

The experimental results show that the clustering of sentence using FUZZY rule work in an efficient manner on considering with the feature extraction based on the processing time and overlapping clusters. A mean and deviance result gives the similarity measures on the basis of hard, soft and medium similarities. On the experimental result, the cluster information obtained gives the number of times the word existence on the given benchmark dataset. This work further can be extended to produce the efficiency of FUZZY clustering on the basis of centrality measures that can be represented in graphical ways. Future research can also deal with hierarchical fuzzy relational Clustering algorithm in an effective manner. In this paper, we studied FRECCA Algorithm and this algorithm is able to achieve superior performance to benchmark Spectral Clustering and k-Medoids algorithms when externally evaluated in hard clustering mode on a challenging data set of famous quotations, and applying the algorithm to a recent news article has demonstrated that the algorithm is capable of identifying overlapping clusters of semantically related sentences. Expectation step calculate Page Rank value for each object in each cluster. Discovered new information in the form of cluster, we can use for market analysis and business treats.

FRECCA has a several number of attractive features. First, based on empirical observations, it is not sensitive to the initialization of cluster membership values, with repeated trials on all data sets converging to exactly the same values, irrespective of initialization. This is in stark contrast to k-Means and Gaussian mixture approaches, which tend to be highly sensitive to initialization. Second, the algorithm appears to be able to converge to an appropriate number o clusters, even if the number of initial clusters was set very high. For example, on the quotations data set the final number of clusters was never greater than eight (there were five actual classes in the data set), and on the news article the algorithm converged to five clusters, which appears reasonable given the length, breadth, and general nature of the article. Finally, while we have applied the algorithm using symmetric similarity measures, the algorithm can also be applied to asymmetric matrices .

REFERENCES

[1]. V. Hatzivassiloglou, J.L. Klavans, M.L. Holcombe, R. Barzilay, M. Kan, and K.R. Mc Keown, “SIMFINDER Flexible Clustering Tool for Summarization,” Proc.ation, pp. 41-49, 2013.

[2]. K. Aas and L. Eikvil, “Text Categorizations: A S, Norwegian Computing Center, 2012.

[3]. H. Zha, “Generic Summarization and Key phrase Extraction Using Mutual Reinforcement Principle and Sentence clustering,” Proc. 5th Ann. Int’l

[4]. R.M. Aliguyev, “A New Sentence Similarity Measure Sentence Based Extractive Technique for Automatic text Summarization,” Expert Systems with Applications, 2011.

[5]. H. Zha, “Generic Summarization and Key phrase Technical Report NR 941, Norwegian by Computer. Addison- Wesley, 2001.and C.

[6]. R. Mihalcea, C. Corley, and C. Strapparava Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity,” Proc. ial Intelligence, pp. 775- 780, 2006.

Uday D Patil – Student of ME Computer Engineering, Jayawantrao Sawant College of Engineering, Hadapsar, Pune University, India.

Prof. Sharmila M. Shinde, Assit. Professor, Head of Computer Engineering Departement, Jayawantrao Sawant College of Engineering, Hadapsar, Pune University, India.