Text Clustering Using Novel Hybrid Algorithm

Divya D. Dev[1] and Merlin Jebaruby[2]

[1] Anna University, Computer Science and Engineering, Chennai-25, India
divyaddev@gmail.com
[2] Vel Tech Hi Tech, Electrical and Communication Engineering, Chennai – 90, India
merlinjebaruby@gmail.com

Abstract. Feature clustering has evolved to be a powerful method for clustering text documents. In this paper we propose a hybrid similarity based clustering algorithm for feature clustering. Documents are represented by keywords. These words are grouped into clusters, based on efficient similarity computations. Documents with related words are grouped into clusters. The clusters are characterised by similarity equations, graph based similarity measures and Gaussian parameters. As words are been given into the system, clusters would be generated automatically. The hybrid mechanism works with membership algorithms to identify documents that match with one another and can be grouped into clusters. The method works to find the real distribution of words in the text documents. Experimental results do show that the proposed method is much better when compared against several other clustering methods. The distinguished clusters are identified by a unique group of top keywords, obtained from the documents of a cluster.

1 Introduction

One of the main segments in text mining would be text clustering or document clustering. This is a process by which documents of similar topics or themes can be grouped together. The clusters can be used to improve the reliability, availability and dimensionality of text mining applications. Effective clustering will make the process of information retrieval and document summarization a lot easier. Document clustering revolves around three major problems. The very first one is on how to identify the similarity between documents. The second issue is on how to decide on the final number of document clusters and the third problem deals with the formation of precise clusters. The concept of feature clustering originates from early methods, which would convert the representation of high dimensional data to lower dimensional data sets. Real time applications have made use of linear algorithms due to its efficient and precise nature. The computational complexity is improved by various algorithms as mentioned in [1][2][3]. Feature clustering is one such algorithm that allows documents with pair wise semantic relatedness to be grouped together. Each document will be identified by a minimal number of features or words; hence the overall dimensionality could be reduced by drastic amounts. The motivation of clustering documents with keywords is done due to two aspects. The keywords can be used to reduce the overall

N.T. Nguyen et al. (Eds.): ACIIDS 2014, Part I, LNAI 8397, pp. 11–20, 2014.

dimensionality of a document set. The traditional methods work with huge bags of words that will increase the complexity of text clustering. Thus, as keywords are been extracted, the documents can be indexed with the top collection of words. This will change the representation of documents and result in sparse text documents. The second phase of clustering makes use of the appropriate keywords. This will increase the comprehensibility of clusters. Additionally, the frequent keyword term set will provide documents with contextual and conceptual meanings.

This paper is organised as follows: Section 2 gives an overview of the existing frequent feature term sets based clustering, this includes DC, Word variance based selection, CFWS, FIHC and Fuzzy clustering. Section 3 proposes a novel method called MMMC, a series of Graph Based Similarity measures and efficient Gaussian Computations to construct relevant documents and form optimized clusters. The proposed method normalizes by merging and splitting clusters in an effective manner. Section 4 showcases the experimental evaluation of Hybrid Clustering. Section 5 concludes the paper.

2 Related Work

2.1 Document Frequency

Document frequency is the count of the number of documents in which a term could occur. It is regarded as an easy measure with straightforward criterion by which large datasets can be grouped together at linear computation complexity. The method is simple and is ideal for effective feature selection [4].

2.2 Word Variance Based Selection

Word variance could be used to differentiate the words of a dataset. The algorithm keeps together words with higher variances [4]. The unique variance of terms is then used to sort the words. The cluster size will be equivalent to the number of documents in it. The variance of word w, in document x, and occurrence x(w) in a dataset with N document sets is defined as:

$$\sigma^2 = \frac{1}{N}\sum_x x^2(w) - \left(\frac{1}{N}\sum_x x(w)\right)^2 \tag{1}$$

2.3 Clustering Frequent Word Sequence

Clustering frequent word sequence is a method of clustering proposed in [5]. The technique uses frequent word sequence and K-mismatch for text clustering. The word sequence method takes into consideration the order in which the words are placed. Though K-mismatch is used to form clusters, the presence of transitivity makes certain that documents appear in more than one cluster. When K-mismatch runs extensively, the clusters with become more ambiguous, as a result all documents will be grouped into one cluster. This is called as trivial clustering.

2.4 Frequent Itemset Based Hierarchical Clustering

Frequent itemset based hierarchical clustering was proposed in [6]. The method describes two kinds of frequent items, namely the cluster frequent item and global frequent item. The hierarchical method works through four phases to produce effective document clusters: the frequent itemsets are found, the initial phase of clustering is performed, trees are constructed and finally pruning is done on the final clusters. FIHC does not deal with pair wise similarity. It clusters using the classic method of clustering. FIHC constructs a similarity matrix from which document pairs with the largest number of similarities will be set as zero and are grouped together.

2.5 Fuzzy Self Clustering Feature Clustering

Fuzzy self-clustering feature clustering algorithm is renowned as an incremental approach. It reduces the dimensionality of documents and groups features that are similar to one another. The clusters are identified by statistical mean and deviation. Words that don't fall into the existent clusters will be placed in newly created clusters.

The proposed method is an extension of my previous work which deals with keyword extraction and is proposed in [8]. The optimal output from D4 Keyword Extractor [8] is passed onto the second phase of Hybrid clustering. Most of the existing "feature clustering" algorithms have few common issues, which affects its net output. First, the users have to mention a value for the desired number of clusters. This is a burden on users. The count has to be indicated by a trial and error method, which has to be repeated manually until an appropriate output is generated. Secondly, the underlying variation in clusters is missed by many algorithms. Variance is an important factor that will calculate the similarity between clusters. Appropriate calculation of variances will result in better data distribution. Thirdly, clusters tend to have the same degree of features. Sometimes, the output will be better if the distribution is uneven and certain clusters are made with a larger number of text documents. The proposed Hybrid algorithm deals with the above problems.

3 Proposed Method

3.1 Strategy 1

The main objective of the Hybrid Algorithm is to produce comprehensive document clusters. The keywords are subject to similarity measures by which the relevance of key terms in a given document will be identified. Initially, a Maximal Must and Minimal Cannot (MMMC) algorithm is proposed in conjuncture with the key terms that will contribute to the document's actual meaning. Thus, MMMC would improve the accuracy of the feature itemsets.

Working Process of Strategy 1

MMMC evaluates the accuracy of the extracted keywords. The strategy identifies the similarity between individual keywords of each document, and would eliminate unpromising keywords based on this estimation. MMMC works as a worst case similarity estimation. It guarantees to produce optimal keywords for a given document. The filtering strategy is quite similar to the top-k selection of Fagin's No Random Access (NRA) algorithm [9]. The keywords, t of a document, d are subject to the equations MM and MC. Terms which satisfy the condition, where maximum scores MM is more than the minimal scores MC will be kept in the collection. Terms that fail to satisfy the relationship of similarity will be removed. In this method of MMMC, the actual score of MM is the similarity between a keyword $w_2(t,d)$ and the document's top most keyword which will act as the document's centroid. MM(t) is defined as:

$$MM\ (t) = \left(\log \left(\frac{1}{\max(\ W\ (t,d) - 1} \right) - w_2(t,d) \right)^2 - w_2(t,d) \qquad (2)$$

Where W(t , d) is a real value that represents the weight of the highest priority keyword. The MC score is a similarity measure that finds a numerical relationship with the least priority keyword and other terms key terms. MC(t) is defined as:

$$MC\ (t) = w_2(t,d) - \sqrt{2 * w_2(t,d) * \log \left(\frac{1}{1 - \min(\ W\ (t,d))} \right)} \qquad (3)$$

The similarity estimation depends on the numerical weight of terms in a document and not all keywords need to be included in the document summary. The computation works in accordance with the following constraints: a) each term will not be included in the summary of a document if it has a MC that is more than its MM, otherwise the term will be a part of the final keyword set b) the number of keywords present in a document must be more than the median count of keywords. If the top keyword does not give an apt length of keywords, it has to be removed and the second top keyword has to be placed as the next prime keyword.

3.2 Strategy 2

The keywords from MMMC will be passed onto a Graph Based Similarity algorithm [10], which will group the keywords to form preliminary clusters. The algorithm works with three formulations that will identify the dependencies between words, through which the centrality and resulting scores of the words in a document can be found. The algorithm covers over three major tasks: a) the top keywords of the document will form the graph's vertices b) graph dependencies will be constructed with the similar words c) the graph edges are assigned labels with promising scores. The dependencies of words in a document can be represented as a graph. Given a documents D= $\{d_1, d_2, d_3, \ldots d_n\}$ with a sequence of keywords W = $\{w_1, w_2, w_3, \ldots w_n\}$. Similar words of different documents can be connected with admissible labels L= $\{l_1, l_2, l_3, \ldots l_n\}$. We define a label for the Graph G = (V,E) when the keyword weight w_1, w_2 satisfies the three measures of similarity equations [5][6][7]. Note, the graph does not need to be fully connected, as the edges will be labelled only if the graph based similarity measures are satisfied. The information of clustering is drawn from the entire graph. The similarity of keywords in a document is defined as:

$$E_1 = -\log \frac{|w_1 - w_2|}{avg\ (w_1, w_2)} \tag{6}$$

$$E_2 = \frac{low\ (w_1, w_2)}{avg\ (w_1, w_2)} \tag{7}$$

$$E_3 = \frac{\log(\ low\ (w_1, w_2))}{\log(\ w_1) + \log(\ w_2)} \tag{8}$$

$$|E_3\text{-}E_2|<=|E_1\text{-}E_2|<=|E_3\text{-}E_1| \tag{9}$$

The word similarity metrics are derived from Word Net-based implementation. Equation (6) is determined by the similarity measure proposed by Leacock & Chohorow [11], where $|w_1\text{-}w_2|$ gives the difference in weight of the two keywords. Equation (7) is formed with the concept proposed by Lesk [12]. The similarity function identifies the overlapping nature of words. The lowest weight is taken to characterise the two different keywords. Equation (8) is a Wu and Palmer [13] similarity metric which uses the depth of two keywords and the depth of the least common subsume.

3.3 Strategy 3

As a result of the Graph Based Similarity Measure, we are given with a document set D of "n" different documents $d_1,d_2,d_3...d_n$ in "p" different clusters C of $c_1,c_2,c_3...c_p$. Each document will have a unique set of keyword to describe it. Using which we can construct an accurate word pattern x_i for each word w_i, with an occurrence of O_{pi}, is quite similar to what is defined in equation [12].

$$w_{ji}=P(c_1|w_i),\ P(c_2|w_i)\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\ P(c_p|w_i) = \tag{10}$$

$$P(c_n|w_i) = \frac{\sum_{n=1}^{p} O_{ni} * \delta_{pk}}{\sum_{n=1}^{p} O_{ni}} \tag{11}$$

It is with these word patterns that the session of optimization would work on. The Hybrid clustering algorithm uses Gaussian parameters for optimization. Once the words, w are grouped into clusters in accordance with its word patterns, each cluster can be characterised by a one dimensional Gaussian Function. Gaussian functions are acknowledged as superior functions in terms of its performance. Thus, let C be a cluster with j word patterns x_j. Let $x_j=<x_1,x_2,x_3....x_j>$, and the standard deviation $\sigma = < \sigma_1, \sigma_2,... \sigma_p>$ of each cluster is defined as:

$$x_j = \frac{\sum_{n=1}^{p} w_{ji}}{|C|} \tag{12}$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{p}(w_{ij} - x_{ij})^2}{|C|}} \tag{13}$$

For every 1<j<p, where |C| represents the size of each cluster, i.e. the total number of word patterns in a given cluster C. Optimization through Gaussian parameters makes use of Fuzzy Similarity. Thus for every cluster with word pattern x_j =< x_1,x_2,...x_j> and standard deviation σ_j =< σ_1, σ... σ_j> a membership function is being represented as:

$$\mu_c(w) = \prod_{i=1}^{p} \exp\left[-\left(\frac{w_i - x_j}{\sigma_i}\right)^2\right] \tag{14}$$

Any word pattern that is similar to its mean value will be a part of the cluster. Thus, words with a membership function output equivalent to one (μ_c(w) ≈ 1)will be a part of the cluster. If a key term has a word pattern that is quite deviant (μ_c(w) ≈ 0) from the cluster's membership function, will hardly be a part of the cluster.

Preliminaries of Strategy 3

Two different cases could occur with the word patterns. Firstly, the word pattern x_i is not similar and it does not fit into the memberships function. Thus, the cluster G_i has to be broken and a new cluster is formed, k=i+1. G_k will have the word pattern x_i, while the word pattern x_i will be removed from G_i. At this point, G_k will have only one word pattern alias word in it. On further iterations, more word patterns could have a membership function which relates to the deviation of x_i. These word patterns can be included in the cluster G_k if and only if the word patterns belonged to documents, which were at least weakly connected with documents of the words in G_k during the Graph Based Similarity Measure. If there is another cluster G_i that produces similar word patterns and membership functions as of G_k, these clusters can be grouped together to form single cluster. Now we will have an optimized number of clusters. These clusters can be stored for future reference. With new training patterns, the algorithm can be run and existing clusters will be modified or new clusters can be created.

4 Experimental Results

In this section, we present the experimental results to show the effectiveness of our hybrid clustering algorithm. Two well known data sets were used to prove our text clustering method: Reuters 21578 and Brown Corpus. Reuters-21578 was obtained from http:kdd.ics.uci.edu/databases/reuters21578/reuters21578. The document collection had 135 different categories. Nevertheless, only 50 different categories were used during the experimentation. Brown Corpus is another data set which contains 500 different samples of text documents. The text documents are distributed around 15 genres. Every word in the document is labelled with part of speech tags. For testing the hybrid algorithm, the complete Brown Corpus document set was used.

4.1 Evaluation Methods

DC, IG, FIHC, CFWS and our Hybrid Clustering algorithm were run on Reuters 21578. The novel method is not compared against traditional methods like k-means and bisecting k-means because the previous methods mentioned in section 2 have been well studied and they are more efficient than the conventional methods. To compare the effectiveness of each method, performance measures in terms of micro-averaged precision (MicroP), micro-averaged recall (MicroR) and micro-averaged F-measure (MicroF1) are used. In the micro averaged formulas, TP_i with respect to each cluster c_i, is the number of documents that are correctly classified to c_i, TN_i is the number of incorrect non-c_i test documents that are classified into the non-c_i clusters. FP_i is the number of non-c_i that is incorrectly classified into c_i. FN_i is the number of false c_i test documents that are classified to non-c_i.

$$MicroP = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p} (TP_i + FP_i)} \qquad MicroR = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p} (TP_i + FN_i)} \qquad MicroF = \frac{2 * MicroP * MicroR}{MicroP + MicroR}$$

4.2 Evaluation Results

Table [1] and Table [2] shows that on Reuters 21578 and Brown corpus, respectively, the novel hybrid algorithm significantly outperforms the other methods used for document clustering.

Table 1. Output for Reuters 21578

No of documents		20	50	80	120	240	500
Microaveraged Precision	FIHC	79.87	79.96	82.82	83.58	83.75	85.38
	WV	91.58	**92.00**	68.82	69.46	76.25	83.74
	DC	81.91	78.35	79.68	79.74	81.42	83.08
	CFWS	85.18	82.96	82.34	82.68	**87.68**	84.73
	Hybrid	**92.34**	91.34	**88.65**	**90.32**	87.28	**87.65**
Microaveraged Recall	FIHC	51.72	53.27	54.93	58.02	61.75	**64.81**
	WV	5.80	11.82	25.72	26.32	36.66	48.16
	DC	49.45	53.05	55.92	56.50	61.32	63.49
	CFWS	**55.33**	61.91	63.41	**63.76**	**65.33**	**66.11**
	Hybrid	54.67	**62.55**	**70.97**	56.67	65.05	60.71
Microaveraged FMeasure	FIHC	62.78	63.94	66.05	68.49	71.09	73.69
	WV	10.91	20.95	37.45	38.17	49.51	61.15
	DC	61.67	63.26	65.72	66.14	69.95	71.98
	CFWS	67.08	70.91	71.65	**72.00**	**74.87**	**74.27**
	Hybrid	**68.68**	**74.25**	**78.85**	69.64	74.54	71.73

Table 2. Output for Brown Corpus

No of documents		20	50	80	120	240	500
Microaveraged Precision	FIHC	88.00	90.97	**91.69**	91.90	92.51	**93.28**
	WV	**91.91**	90.77	89.78	89.07	90.29	89.57
	DC	52.69	56.15	50.12	50.73	48.79	50.54
	CFWS	70.32	78.75	74.45	70.85	62.54	67.89
	Hybrid	82.34	**93.29**	90.40	**92.09**	**93.45**	91.45
Microaveraged Recall	FIHC	62.88	70.80	74.38	73.76	77.54	77.91
	WV	17.75	27.11	30.22	34.16	43.72	52.96
	DC	65.21	74.67	74.96	74.27	**79.84**	**80.61**
	CFWS	70.25	**77.71**	78.32	77.76	78.17	78.11
	Hybrid	**84.32**	77.21	**87.72**	**83.43**	74.55	80.01
Microaveraged FMeasure	FIHC	73.34	79.62	82.13	81.83	**84.36**	84.90
	WV	29.75	41.75	45.21	49.38	58.91	66.56
	DC	58.28	64.09	60.07	60.28	60.56	62.12
	CFWS	70.28	78.23	76.34	74.14	69.49	72.64
	Hybrid	**83.31**	**84.49**	**89.04**	**87.55**	82.94	**85.35**

FIHC outperformed many other methods other than the novel hybrid algorithm. Study showed that FIHC showed favourable amount of performance because it makes use of matching frequent item sets to identify the relationship between clusters and the documents. Nevertheless, FIHC makes use of limited variation. This is because of the method used to form the initial clusters. The very first clusters formed are more skewed i.e. the number of documents present in each cluster varies by considerable amounts, where some clusters have more documents than the others. The difference in terms of documents present in big clusters and the smaller ones is considerably huge. As a result, newer documents introduced into the algorithm are more likely to end in the largest cluster. This is a drawback avoided by the novel hybrid algorithm, where the cluster size does not affect the positioning of documents into clusters. DC and Word Variance Based Clustering failed in most cases because clusters are not formed with the relevance of words in the document but it is based with the number of times a term occurred within the distribution. Thus the clusters are not characterised naturally. To be more precise, the method does not characterise clusters based on the meaningful content of the document. CFWS is another method prone to produce skewed clusters. This is because it uses K-mismatch, which produced very big overlapping between clusters. Thus, the overlapping coefficient in large clusters will always be bigger than what is present in smaller clusters.

5 Conclusion

This paper deals with a hybrid clustering technique that makes use of an incremental approach. The technique reduces document dimensionality and promotes text clustering in a simplified manner. Documents with similar features are grouped together in to a single cluster. Functions based on Gaussian parameters are used to group similar content. This includes an equation to evaluate the word pattern, standard deviation and membership function. If a word does not fall within a given cluster, a new cluster is created for that word. The word pattern and standard deviation of terms are modified automatically, as a new word is positioned into a cluster. The hybrid algorithm works without the help of manual intervention. The desired number of clusters is generated automatically. Furthermore, every cluster is represented by a weighted combination of words. The algorithm uses membership functions to match documents closely. As mentioned previously, the user does not have to mention the number of clusters or the number of documents in each cluster. Thus, errors caused by trial are not present in the hybrid algorithm. Experiments on two different real time data sets proved the effectiveness of our algorithm. On the overall view, the Hybrid Algorithm has a better Microaveraged FMeasure than the other methods. Even as the number of words in each document increased, the hybrid algorithm maintained a better performance. Thus, from the statistical values it is evident that the hybrid algorithm runs better that the existing feature extraction methods.

References

1. Yan, J., Zhang, B., Liu, N., Yan, S., Cheng, Q., Fan, W., Yang, Q., Xi, W., Chen, Z.: Effective And Efficient Dimentionality Reduction For Large Scale And Streaming Data Preprocessing. IEEE Trans. Knowledge and Data Eng. 18(3), 320–333 (2006)
2. Hiraoka, K., Hidai, K., Hamahira, M., Mizoguchi, H., Mishima, T., Yoshizawa, S.: Successive Learning of Linear Discriminat Analysis: Sanger-Type Algorithm. In: Proceedings of IEEE CS Int'l Conf. Pattern Recognition, pp. 2664–2667 (2000)
3. Weng, J., Chang, Y., Hwang, W.S.: Candid Covariance-Free Incremental Principal Component Analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 25(8), 1034–1040 (2003)
4. Yang, Y., Pederson, J.O.: A comparative study on feature selection in text categorization. In: Proc. of the Fourth International Conference on Machine Learning, pp. 412–420 (2007)
5. Li, Y.J., Chungm, S.M., Holt, J.D.: Text document clustering based on frequent word meaning sequences. Data & Knowledge Engineering 64, 381–404 (2008)
6. Fung, B., Wang, K., Ester, M.: Hierarchial Document Clustering Using Frequent Itemsets. In: Proc. of 3rd SIAM International Conference on Data Mining (2003)
7. Khalessizadeh, S.M., Zaefarian, R., Nasseri, S.H., Ardil, E.: Genetic Mining Using Genetic Algorithm For Topic Based On Concept Distribution. World Academy of Science, Engineering and Technology 13 (2006)
8. Rose, J.D., Dev, D.D., Robin, C.R.R.: An Improved Genetic Based Keyword Extraction Technique. In: Terrazas, G., Otero, F.E.B., Masegosa, A.D. (eds.) NICSO 2013. SCI, vol. 512, pp. 153–166. Springer, Heidelberg (2014)
9. Fagin, R., Lotem, A., Naor, M.: Optimal Aggregation Algorithms for Middleware. In: Proc. of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 102–113 (2001)
10. Sinha, R., Mihang, R.: Unsupervised Graph Based Word Sense Disambiguation Using Measure Of Word Semantic Similarity. In: IEEE International Conference on Semantic Computing, pp. 363–369 (2007)

11. Leacock, C., Chodorow, M.: Combining Local Context And Wordnet Sense Similatity For Word Sence Identification In Wordnet, An Electronic Lexical Database. The MIT Press (1998)
12. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: Proc. of the 32nd Annual Meeting of the Association For Computational Linguistics, Las Cruces Mexico (1994)
13. Jiang, J.Y., Liou, R.J., Lee, S.J.: A Fuzzy Self Constructing Feature Clustering Algorithm For Text Classification. IEEE Transactions on Knowledge and Data Engineering 23(3), 335–348 (2011)