ISSN:2229-6093

G Manimekalai et al, Int. J. Comp. Tech. Appl., Vol 2 (5), 1534-1539

A Survey on Various Approaches in Document Clustering

K.Sathiyakumari,
V.Preamsudha,
Assistant professors,
PSGR Krishnammal College for Women,
Coimbatore, India.
aathithi@gmail.com
Preamsudha@gmail.com

G.Manimekalai,
M.Phil Scholar,
PSGR Krishnammal College for Women,
Coimbatore, India.
Mekalawin.mphil@gmail.com

Abstract

Document clustering is the process of segmenting a particular collection of texts into subgroups including content based similar ones. The purpose of document clustering is to meet human interests in information searching and understanding. Nowadays all paper documents are in electronic form, because of quick access and smaller storage. So, it is a major issue to retrieve relevant documents from the larger database. Text mining is not a stand-alone task that human analysts typically engage in. The goal is to transform text composed of everyday language in a structured, database format. In this way, heterogeneous documents are summarized and presented in a uniform manner. Among others, the challenging problems of document clustering are big volume, high dimensionality and complex semantics.

Keywords: text mining, document clustering, information extraction

I. Introduction

The goal of this survey is to provide a comprehensive review of different clustering techniques in text mining. Clustering is the process of organizing data objects into a set of disjoint classes called clusters. Objects that are in the same cluster are similar among themselves and dissimilar to the objects belonging to other clusters. Document clustering is the task of automatically organizing text documents into meaningful clusters or group, In other words, the documents in one cluster share the same topic, and the documents in different clusters represent different topics.

Clustering is an example of unsupervised classification. Classification refers to a procedure that assigns data objects to a set of classes. Unsupervised means clustering does not depends on predefined classes and training examples while classifying the data objects.

Clustering is a crucial area of research, which finds applications in many fields including bioinformatics, pattern recognition, image processing, marketing, data mining, economics, etc. Cluster analysis is one of the primary data analysis tools in data mining. Clustering algorithms are mainly divided into two categories: Hierarchical algorithms and Partition algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step.

Hierarchical algorithms create decomposition of the database D. It is categorized into agglomerative and divisive clustering. Hierarchical clustering builds a tree of clusters, also known as a dendrogram. Every cluster node contains the child cluster. An agglomerative clustering start with a one-point (singleton) cluster and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits into the most appropriate clusters. The process continues until a stopping criterion is achieved. There are two main issues in clustering techniques. At first, finding the optimal number of clusters in a given dataset and secondly, given two sets of clusters, computing a relative measure of goodness between them. For both

these purposes, a criterion function or a validation function is usually applied. In conventional clustering objects that are similar are allocated to the same cluster while objects differ are put in different clusters. These clusters are hard clusters. In soft clustering an object may be in more than two or more clusters.

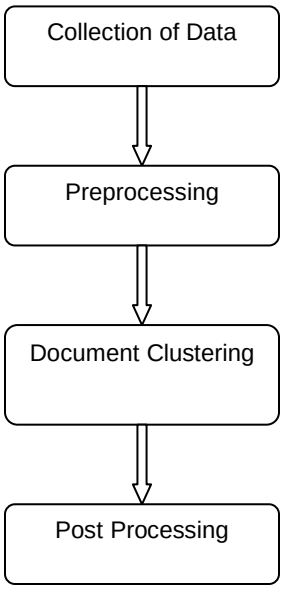

Fig 1: The Stages of the Process of Clustering

Collection of Data includes the processes like crawling, indexing, filtering etc which are used to collect the documents that need to be clustered, index them to store and retrieve in a better way, and filter them to remove the extra data, for example, stop words.

Preprocessing consists of steps that take as input a plain text document and output a set of tokens (which can be single terms or n-grams) to be included in the vector model. These steps typically consist of:

Filtering is the process of removing special characters and punctuation that are not thought to hold any discriminative power under the vector model. This is more critical in the case of formatted documents, such as web pages, where formatting tags can either be discarded or identified and their constituent terms attributed different weights [4].

Tokenization splits sentences into individual tokens, typically words. More sophisticated methods, drawn from the field of NLP, parse the grammatical structure of the text to pick significant terms or chunks, such as noun phrases [5].

Stemming the process of reducing words to their base form, or stem. For example, the words “connected", “connection", “connections" are all reduced to the stem “connect." Porter's algorithm [6] is the de facto standard stemming algorithm.

Stopword removal A stopword is defined as a term, which is not thought to convey any meaning as a dimension in the vector space (i.e. without context). A typical method to remove stopwords is to compare each term with a compilation of known stopwords. Another approach is to first apply a part-of-speech tagger and then reject all tokens that are not nouns, verbs, or adjectives.

Pruning removes words that appear with very low frequency throughout the corpus. The underlying assumption is that these words, even if they had any discriminating power, would form too small clusters to be useful. A pre-specified threshold is typically used, e.g. a small fraction of the number of words in the corpus. Sometimes words which occur too frequently (e.g. in 40% or more of the documents) are also removed.

Document Clustering is the main focus of this paper and will be discussed in detail.

Post processing includes the major applications in which the document clustering is used, for example, the application that uses the results of clustering for recommending news articles to the users.

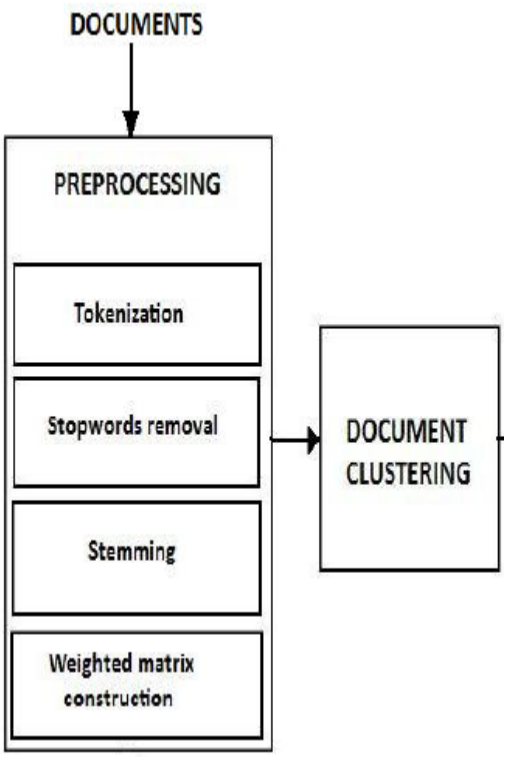

Fig 2: Stages of Document Clustering

The problem of document clustering is generally defined as follows [1] given a set of document clustering. An automatically derived number of clusters, such that the documents assigned to each cluster are more similar to each other than the documents assigned to different clusters. Documents are represented using the vector space model that treats a document as a bag of words [2]. A major characteristic of document clustering algorithms is the high dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. They could not work efficiently in high dimensional feature spaces due to the inherent sparseness of the data. The next challenge is that not all features are important for document clustering, some of the features may be redundant or irrelevant and some may even misguide the clustering result [3], especially there are more irrelevant more features than relevant ones.

The remainder of this paper is organized as follows. Section II discusses some of the earlier related work on document clustering. Section III provides a fundamental idea on which the future research work and conclusion.

II Related Works

In this subsection, we review some of the clustering techniques related to this study.

2.1 Partitional Clustering

The partitional or non-hierarchical document clustering approaches attempt a flat partitioning of a collection of documents into a predefined number of disjoint clusters. Partitional clustering algorithms are divided into iterative or reallocation methods and single pass methods. Most of them are iterative and the single pass methods are usually used in the beginning of a reallocation method, in order to produce the first partitioning of the data.

Partitional clustering algorithms compute a *k*-way clustering of a set of documents either directly or via a sequence of repeated bisections. A direct *k*-way clustering is commonly computed as follows. Initially, a set of *k* documents is selected from the collection to act as the *seeds* of the *k* clusters. Then, for each document, its similarity to these *k* seeds is computed, and it is assigned to the cluster corresponding to its most similar seed. This forms the initial *k*-way clustering. This clustering is then repeatedly refined so that it optimizes the desired clustering criterion function. A *k*-way partitioning via repeated bisections is obtained by recursively applying the above algorithm to compute 2-way clustering (*i.e.*, Bisections). Initially, the documents are partitioned into two clusters, and then one of these clusters is selected and is further bisected, and so on. This process continues *k* -1 times, leading to *k* clusters. Each of these bisections is performed so that the resulting two-way clustering solution optimizes the particular criterion function.

M. Steinbach, G. Karypis, and V. Kumar [9] the overall *k*-way clustering solution will not necessarily be at local minima with respect to the criterion function. The key step in this algorithm is the method used to select which cluster to bisect next. In all of our experiments, we chose to select the largest cluster, as this approach lead to reasonably good and balanced clustering solutions [9]. Extensive experiments presented in [10], show that the clustering solutions obtained via repeated bisections are comparable or better than those produced via direct clustering. Furthermore, their computational requirements

are much smaller, as they have to solve a simple optimization problem at each step. For this reason, in all of our experiments we use this approach to compute partitional clustering solutions.

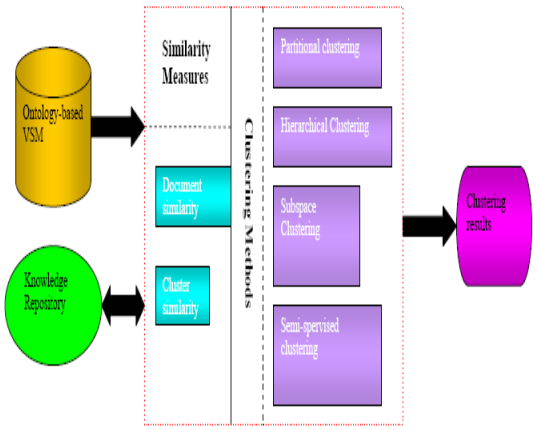

Fig 3: Clustering Methods

Y. Zhao, G. Karypis [10] One of the differences between partitional and agglomerative clustering algorithms is the fact that the former do not generate an agglomerative tree. Agglomerative trees are very useful as they contain information on how the different documents are related to each other, at different levels of granularity. One-way of inducing an agglomerative tree from a partitional clustering solution is to do it in such a way so that it preserves the already computed *k*-way clustering. This can be done in two steps. First, we build an agglomerative tree for the documents belonging to each one of the clusters, and then we combine these trees by building an agglomerative tree, whose leaves are the partitionally discovered clusters. This approach ensures that the *k*-way clustering solution induced by the overall tree is identical to the *k*-way clustering solution computed by the partitional algorithm. Both of these trees are constructed so that they optimize the same criterion function that was used to derive the partitional clustering solution.

2.2 Hierarchical Clustering Algorithms

Hierarchical clustering approaches attempt to create a hierarchical decomposition of the given document collection thus achieving a hierarchical structure. Hierarchical methods are usually classified into Agglomerative and Divisive methods depending on how the hierarchy is constructed.

Agglomerative methods start with an initial clustering of the term space, where all documents are considered to represent a separate cluster. The closest clusters using a given inter-cluster similarity measure are then merged continuously until only 1 cluster or a predefined number of clusters remain.

$$\vec{\mu}(C) = \frac{1}{|C|}\sum_{\vec{x}\in C}\vec{x}$$

Simple Agglomerative Clustering Algorithms are

1. Compute the similarity between all pairs of clusters i.e. calculates a similarity matrix whose ijth entry gives the similarity between the ith and jth clusters.
2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pairwise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

Divisive clustering algorithms start with a single cluster containing all documents. It then continuously divides clusters until all documents are contained in their own cluster or a redefined number of clusters are found.

Agglomerative algorithms are usually classified according to the inter-cluster similarity measure they use. The most popular of these are single-link, complete-link and group average. In the single link method, the distance between clusters is the minimum distance between any pair of elements drawn from these clusters (one from each), in the complete link it is the maximum distance and in the average *link* it is correspondingly an average distance.

2.2.1 Advantages of hierarchical clustering

1. Embedded flexibility regarding the level of granularity.
2. Ease of handling any forms of similarity or distance.
3. Applicability to any attributes type.

2.2.2 Disadvantages of hierarchical clustering

1. Vagueness of termination criteria.
2. Most hierarchal algorithm does not revisit once constructed clusters with the purpose of improvement.

2.3 K-Means

K-means is the most important flat clustering algorithm. The objective function of K means is to minimize the average squared distance of objects from their cluster centers, where a cluster center is defined as the mean or centroid μ of the objects in a cluster C:

The ideal cluster in K-means is a sphere with the centroid as its center of gravity. Ideally, the clusters should not overlap. A measure of how well the centroids represent the members of their clusters is the Residual Sum of Squares (RSS), the squared distance of each vector from its centroid summed over all vectors

$$RSS_i = \sum_{\vec{x} \in C_i} \| \vec{x} - \vec{\mu}(C_i) \|^2$$

$$RSS = \sum_{i=1}^{K} RSS_i$$

K-means can start by selecting as initial cluster centers K randomly chosen objects, namely the seeds. It then moves the cluster centers around in space in order to minimize RSS. This is done iteratively by repeating two steps until a stopping criterion is met

1. Reassigning objects in the cluster with the closest centroid.
2. recomputing each centroid based on the current members of its cluster.

We can use one of the following terms conditions as stopping criterion

1. A fixed number of iterations have been completed.
2. Centroids μi do not change between iterations.
3. Terminate when RSS falls below a pre-established threshold.

Algorithm for K-Means

```
procedure KMEANS(X,K)
    {s1, s2, · · · , sk}  SelectRandomSeeds(K,X)
    for i ←1,K do
        μ(Ci) ← si
    end for
    repeat
        min_{k~xn-~μ(Ck)k} Ck = Ck [ {~xn}
        for all Ck do
            μ(Ck) = 1
        end for
    until stopping criterion is met
end procedure
```

2.4 Expectation Maximization

The EM algorithm fall within a subcategory of the flat clustering algorithms, called Model-based clustering. The model-based clustering assumes that data were generated by a model and then tries to recover the original model from the data. This model then defines clusters and the cluster membership of data.

The EM algorithm is a generalization of K-Means algorithm in which the set of K centroids as the model that generate the data. It alternates between an expectation step, corresponding to reassignment, and a maximization step, corresponding to the recomputation of the parameters of the model.

III.FUTURE WORK AND CONCLUSION

In this survey we had projected various clustering approaches and algorithms in document clustering .The area of document clustering have many issues, which need to be solved. We hope, the paper gives interested readers a broad overview of the existing techniques. As a future work, improvement

over the existing systems with better results which offer new information representation capabilities with different techniques like search result clustering, collection clustering and co-clustering can be attempted.

References

[1] Prof. K. Raja, C. Prakash Narayanan, "Clustering Technique with Feature Selection for Text Documents", Proceedings of the Int.Conf. on Information Science and Applications ICISA 2010 6 February 2010, Chennai, India.

[2] Luiz G. P. Almeida, Ana T. R. Vasconcelos and Marco A. G. Maia," A Simple and Fast Term Selection Procedure for Text Clustering "Seventh International Conference on Intelligent Systems Design and Applications, 0-7695-2976-3/07 © 2007 IEEE, doi:10.1109/ISDA.2007.15

[3] Fabrizio Sebastiani "Machine Learning in Automated Text Categorization" ACM Computing Surveys, Vol. 34, No. 1, March 2002

[4]K. M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. IEEE Transactions on knowledge and data engineering, 16(10):1279{1296, 2004.

[5]George A. Miller. Wordnet: a lexical database for English. Common. ACM, 38(11):39{41, 1995.

[6] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In ECIR '04: 27th European conference on IR research, pages 181{196, Sunderland, UK, April 2004.

[7]M. F. Porter. An algorithm for suffix stripping. Program, 14(3):130{137, 1980.

[8]A. Hotho S. Staab and G. Stumme. Wordnet improves text document clustering. In the Semantic Web Workshop at SIGIR 2003, 26th annual international ACM SIGIR conference, July 2003.

[9] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000.

[10] Y. Zhao, G. Karypis. Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report #01-34, University of Minnesota, MN 2001.

[11] Chen Wenliang, Chang Xingzhi, and Wang Huizhen, "Automatic Word Clustering for Text Categorization Using Global Information" Copyright ACM 2004.

[12] Dino Ienco Rosa Meo "Exploration and Reduction of the Feature Space by Hierarchical Clustering" Dipartimento di Informatica, Universit`a di Torino, Italy.

[13] Fabrizio Sebastiani "Machine Learning in Automated Text Categorization" ACM Computing Surveys, Vol. 34, No. 1, March 2002.

[14] George Forman "Feature Selection: We've barely scratched the surface" Published in IEEE Intelligent Systems, November 2005.

[15] Hisham Al-Mubaid and Syed A. Umair "A New Text Categorization Technique Using Distributional Clustering and Learning Logic" IEEE 2006.

[16] Huan Liu and Lei Yu "Toward Integrating Feature Selection Algorithms for Classification and Clustering" Department of Computer Science and Engineering,2005.

[17] Huan Liu, "Evolving Feature Selection" Published by the IEEE Computer Society 2003.

[18] Jinxiu Chen1 Donghong Ji1 Chew Lim Tan "Unsupervised Feature Selection for Relation Extraction" Institute for Infocomm Research 2002.

[19] Martin H.C. Law and Mario A.T. Figueiredo "Simultaneous Feature Selection and Clustering Using Mixture Models" iiIEEE 2004.

[20] Tao Liu and Shengping Liu "An Evaluation on Feature Selection for Text Clustering" Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[21] Yanjun Li Congnan Luo, "Text Clustering with Feature Selection by Using Statistical Data" IEEE 2008.