# Synopsis for Project
# On
# Document Clustering

Pankaj Jajoo
03CS3024

Under the Guidance of

**Prof. Sudeshna Sarkar**
*Professor*
*Computer Science & Engineering Department*
Indian Institute of Technology Kharagpur
WB, India 721302

# Introduction

Automatic clustering of documents is an increasingly important tool for handling the exponential growth in available online texts. Many algorithms have been suggested for this task in the past few years. The most common approaches start by evaluating the co-occurrence matrix of words versus documents, given document training data. This is a *supervised classification* of documents. It is difficult to find data for training and also it is not time efficient.

For *unsupervised classification* of documents the common approach used is the K-Means clustering approach in which the words in a document are taken as features of the document and then they are clustered based on common features. In case of a large data set the documents are represented in high-dimensional sparse feature space which results in a lot of noise.

In this project, we tried to improve the clustering results by applying two algorithms:

1. Feature based Clustering
2. Triplet based graph partitioning

In Feature based clustering we first cluster the features of the documents, which are the words in this case, by creating a graph of the words based on their co-occurrence in the documents, and then we cluster the documents based on the word clusters.

In Triplet based graph partitioning, we first create a graph of the documents with edge weight as similarity between the documents. Then by taking the triplets with high edge weights as the base we preprocess the graph such that there are more edges among the nodes which should be in same clusters and few edges between nodes which should be in separate clusters. Then we apply a standard graph clustering algorithm to cluster this graph.

# Motivation

Document clustering has been used in many different areas of text mining and information retrieval. Initially it was used for improving the precision and recall in

information retrieval systems and finding nearest neighbors of a document. Later it has also been used for organizing the results returned by a search engine and generating hierarchical clusters of documents.

Initially we applied the K-Means and Agglomerative Hierarchical clustering methods on the data and found that the results were not very satisfactory and the main reason for this was the noise in the graph, created for the data. This provided us the motivation for trying a pre-processing of the graph to remove the extra edges. We applied a heuristic for removing the inter cluster edges and then applied the standard graph clustering methods to get much better results.

We also tried a completely different approach by first clustering the words of the documents by using a standard clustering approach and thus reducing the noise and then using this word cluster to cluster the documents. We found that this also gave better results than the classical K-Means and Agglomerative Hierarchical clustering methods.

## Related Work

The two commonly used techniques for document clustering are K-Means and Agglomerative Hierarchical clustering.

The basic idea of K-Means approach is to select K random documents as the centroids of K required clusters and assign all other documents to these centroids. Then re-evaluate the centroids of the clusters formed and repeat the above procedure until the centroids do not change.

In agglomerative hierarchical clustering approach each document is considered as a separate cluster and then at each time step two most similar clusters are merged together to reduce a cluster. This is done unless required numbers of clusters are left.

The evaluation of these techniques is done in two standard ways:

1. Internal quality measure
2. External Quality measure

In internal quality measures, the overall similarity measure is used based on the pair wise similarity of documents and there is no external knowledge to be used.

For external quality measure some external knowledge for the data is required. In this case the external knowledge is the pre-assigned categories of the documents. Entropy is the standard measure of quality of the clusters. For each cluster, the category distribution of data is calculated first i.e $p_{ij}$ is the probability that a member of cluster j belongs to category i. Then the entropy of each cluster j is calculated as

$$E_j = -\sum_i p_{ij} \log(p_{ij})$$

The total entropy is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{en} = \sum_{j=1}^{m} \frac{n_j * E_j}{n}$$

Where m is the total number of clusters, $n_j$ is the size of $j^{th}$ cluster and n is the total number of documents

# Work done

## 1. *Feature based Clustering*

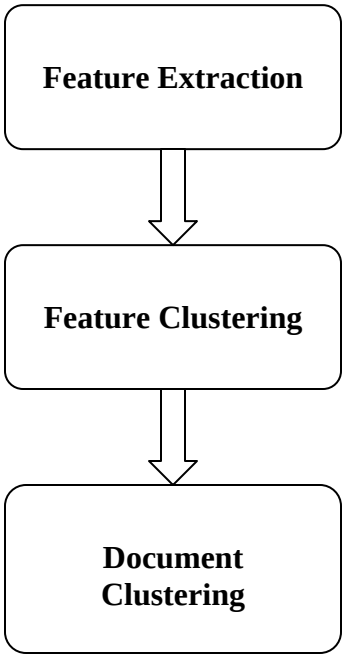


**Figure 1. Feature based Clustering Approach used for Clustering of Documents**

### Feature Extraction

This is used for extraction of features( important words and phrases in this case) from the documents. We have used Named-Entity tagger and frequency of unigrams and bigrams to extract the important words from the documents.

### Feature Clustering

This is the most important phase in which the extracted features are clustered based on their co-occurrence. For this we tried many algorithms and found Multi-level graph clustering algorithms to be best for large data set as it reduces the time taken to a large extent.

## Document Clustering

This is the final phase in which documents are clustered using the feature clusters. For this we have used a simple approach in which a document is assigned to the cluster of words of which it has the maximum words.

## Multi-level Graph Partitioning Algorithm

The multilevel algorithms are graph clustering algorithms which take a graph as input in which an edge defines the similarity between two nodes it is connecting. Based on these similarities it clusters the nodes.
The overview of a multilevel algorithm is this:

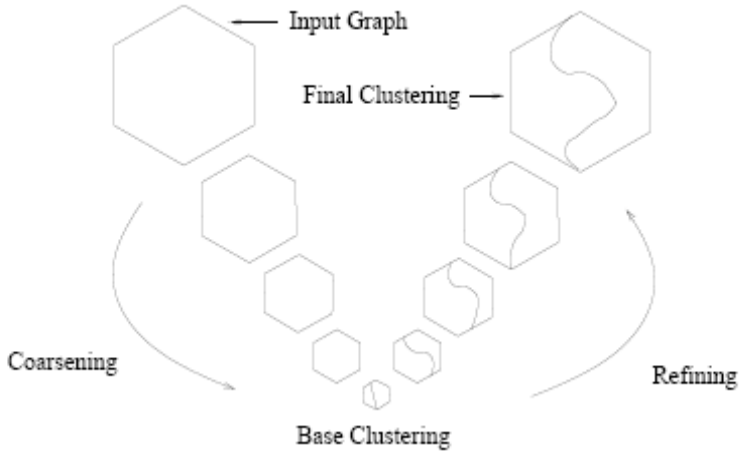


**Figure 2. Multi-level graph partitioning algorithm**

The three phases are:

1. *Coarsening*

   In the coarsening phase the graph is repeatedly transformed into smaller and smaller graphs by combining set of nodes to form supernodes. When combining a set of nodes into a single supernode, the edge weights out of the supernode are taken to be the sum of the edge weights out of the original nodes.

2. *Base Clustering*

   The graph is coarsened until it becomes small enough to be clustered easily and effectively. At this point base clustering is performed by directly clustering the coarsened graph. The algorithms used for base clustering are the usual graph clustering algorithms.

3. *Refining*

   In the refinement phase the clustered initial graph is gained by separating the nodes which were combined in the coarsening phase. Given graph $G_i$ , the graph $G_{i-1}$ is obtained which is the graph used in level i-1 of the coarsening phase.

   The clustering in $G_i$ induces a clustering in $G_{i-1}$ as follows:

   if a supernode in $G_i$ is in cluster c, then all nodes in $G_{i-1}$ formed from that supernode are in cluster c. This yields an initial clustering for the graph $G_{i-1}$, which is then improved using a refinement algorithm.

## *2. Triplet based graph partitioning*

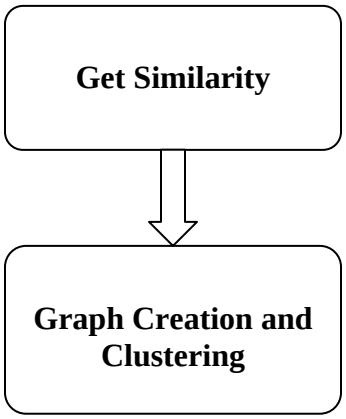


**Figure 3. Triplet based graph partitioning algorithm**

## Get Similarity

Similarity between every pair of documents is calculated.

For any two documents X and Y, where X and Y are the sets of unique words in the documents except the stopwords, the similarity is defined as :

$$\text{Sim}(X,Y) = \frac{|X \bigcap Y|}{\min.(|X|,|Y|)}$$

## Graph Creation and Clustering

This is the most important part of the algorithm. In this, a graph is created with every document as a node and the edges are drawn using following algorithm:

1. Draw an edge between two documents, i and j, if there exists a third document, k , such that Sim(i,k) >= *Threshold* and Sim(j,k) >= *Threshold*, where *Threshold* is a value between 0 and 1.
2. Take Sim(i,j) as the edge weight.
3. Now, in this graph, keep an edge between two nodes, i and j, if there exists a third node k, such that edge-weight(i,k) >= *Threshold* and edge-weight(k,j) >= *Threshold* and edge-weight(i,j) > 0.
4. Cluster the above graph, with each edge considered equal weighted, using a standard graph clustering algorithm.
5. If some documents remain then reapply the algorithm with a lower value of *Threshold* and each created cluster as a node.
6. Merge the clusters if less number of clusters are required.

The pre-processing of the graph has been done inorder to reduce the noise and density of the graph by removing the edges heuristically.

Here two graph clustering algorithms have been used –

1. Multilevel graph partitioning algorithm described above.
2. Marcov Clustering Algorithm (MCL)

# Results

We have used two data sets for evaluation:

1. 20 newsgroups data which contains around 20,000 news articles categorized in 20 categories.
2. A keepmedia news article set of around 62,000 news articles categorized in 69 categories.

The preliminary motivating results for a small dataset are:

1. 20 newsgroups data

   Number of articles = 1500

   Number of clusters = 100

   Entropy:

   a. K-Means approach - 1.65
   b. Feature based clustering – 0.828
   c. Triplet based graph partitioning – 0.45

2. Keepmedia data

   Number of articles = 1500

   Number of clusters = 100

   Entropy:

   a. K-Means approach – 1.89
   b. Feature based clustering – 0.536
   c. Triplet based graph partitioning – 0.72

# References


**[1]** Noam Slonim and Naftali Tishby.
"*The Power of Word Clusters for Text Classification*"
School of Computer Science and Engineering and The Interdisciplinary Center for Neural Computation The Hebrew University, Jerusalem 91904, Israel

**[2]** Noam Slonim and Naftali Tishby.
"*Document Clustering using Word Clusters via the Information Bottleneck method*"
School of Computer Science and Engineering and The Interdisciplinary Center for Neural Computation The Hebrew University, Jerusalem 91904, Israel

**[3]** Inderjit S. Dhillon, *Member, IEEE*, Yuqiang Guan and Brian Kulis
"*Weighted Graph Cuts without Eigenvectors: A Multilevel Approach*"

**[4]** Michael Steinbach, George Karypis, and Vipin Kumar.
"*A Comparison of Document Clustering Techniques*"
Department of Computer Science and Engineering, University of Minnesota

**[5]** Anton V. Leouski and W. Bruce Croft
"*An Evaluation of techniques for clustering search results*"
Computer Science Department, University of Massachusetts at Amherst