International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 485 - 489

Text Categorization of Documents using K-Means and K-Means++ Clustering Algorithm

Aditi Anand Shetkar
Department of Information Technology
Padre Conceicao College of Engineering
Verna-Goa, India
shetkar.aditi29@gmail.com

Sonia Fernandes
Department of Information Technology
Padre Conceicao College of Engineering
Verna-Goa, India
fdezsonia@gmail.com

***Abstract*—** Text categorization is the technique used for sorting a set of documents into categories from a predefined set. Text categorization is useful in better management and retrieval of the text documents and also makes document retrieval as a simple task. Clustering is an unsupervised learning technique aimed at grouping a set of objects into clusters. Text document Clustering means clustering of related text documents into groups based upon their content. Various clustering algorithms are available for text categorization.
This paper presents categorization of the text documents using two clustering algorithms namely K-means and K-means++ and a comparison is carried out to find which algorithm is best for categorizing text documents. This project also introduces pre-processing phase, which in turn includes tokenization, stop-words removal and stemming. It also involves Tf-Idf calculation. In addition, the impact of the three distance/similarity measures (Cosine Similarity, Jaccard coefficient, Euclidean distance) on the results of both clustering algorithms(K-means and K-means++) are evaluated.
The dataset considered for evaluation consists of 600 text documents of three different categories- Festivals, Sports and Tourism in India. Our observation shows that for categorizing the text documents using K-Means++ clustering algorithm with Cosine Similarity measure gives better result as compared to K-means. For K-Means++ algorithm using Cosine Similarity measure purity of the cluster obtained is 0.8216.

Keywords-*Text Categorization,Document Clustering,K-Means,K-Means++,Cosine Similarity*

*****

I. INTRODUCTION

Data mining is the study of the KDD also known as "knowledge discovery in databases" process. The main objective of data mining is to extract patterns and knowledge from huge collection of data, not the extraction of data itself. Information retrieval (IR) is the activity of finding information resources trivial to an information need from a collection of data resources. Searches are based on full-text indexing. Text categorization is an Information Retrieval task and Clustering is Data Mining method.

For making retrieval of the text document simple and easy task, there is a need to do text categorization of documents automatically. Several algorithms are available for text categorization. One approach is automatically grouping of text documents using clustering algorithms. Several clustering algorithms are also available for text categorization.

Text document clustering includes the utilization of descriptors and descriptor extraction. Descriptors are sets of words that portray the substance inside the cluster. Text document clustering is by and large thought to be a unified procedure.

In this paper, the plain text documents are clustered based on two clustering algorithms namely K-Means and K-Means++.This two algorithms are then compared to see which algorithm is best suited for text clustering.

The keyword "categorization" in our paper refers to the assignment of documents to categories, whereas "clustering" is grouping of documents without recognizing the meaning of groups/clusters. The groups/clusters are named by a set of keywords that describes the similar content of the documents in the group/cluster.

The rest of the paper is proposed as follows. Section 2 describes the text clustering techniques; Section 3 outlines the proposed work and system architecture; Section 4 presents experimental setup and observations; Section 5 presents conclusion and future scope.

II. TEXT CLUSTERING APPROACHES

In this section, two partitioning algorithms namely K-Means and K-Means++ are described.

A. K-means Clustering Algorithm

The k-means algorithm (MacQueen, 1967) is an unsupervised learning method. The alphabet "K" in the K-means algorithm refers to the number of clusters (groups) we want to assign in the given dataset. If "N" data objects have to be grouped into "K" clusters centers have to be initialized. Each object is then assigned to its closest cluster center and the center of the cluster is updated until the state of no change in each cluster center is reached.

Input:
K: the number of clusters,
Output:
A fix set of k clusters.
Method:
Step 1: Select k numbers of clusters to be determined.
Step 2: Select c_k centroids randomly as the initial centers of the clusters.
Step 3: Repeat
  3.1: Assign each object to their nearby cluster center using similarity measure/distance.
  3.2: Compute new cluster center by calculating mean points.
Step 4: Until
  4.1: No conversion in cluster center OR
  4.2: No object changes its clusters.
Advantages of K-Means Algorithm

- Algorithm is very simple and can be easily implemented.

- Speed is faster which allows this K-Means algorithm to run on large datasets.

Disadvantages of K-Means Algorithm

- Algorithm gives different results on different runs.
- Chooses initial cluster centers randomly which gives bad results.
- Unable to handle outliers and noisy data.

B. K-means++ Clustering Algorithm

K-means++ (David Arthur et. Al., 2007) is another variation of K-means, a new methodology to select initial cluster centers by random starting centers with definite probabilities is used. The steps of K-Means++ algorithm are described below in detail:

Step 1: Given the dataset X, selection of first initial cluster center c1 is done randomly.

Step 2: Select next cluster center ci, selecting ci =x' ∈X with probability pi where $D(x')^2 / \sum_{x\in X} D(x)^2$, represents the shortest distance from x to the close by center already selected.

Step 3: Repeat step 2 till the selection of k cluster centers is done.

Step 4: Once initial k cluster centers are chosen, to get final k clusters apply K-Means Algorithm.

Advantages of K-Means++ Algorithm

- K-Means++ extends the K-Means algorithm by choosing the initial cluster centers according to the probability metrics, and not uniformly at random from the data.
- Faster, robust, easy to understand.
- Consistently finds a better cluster with a lower potential than K-Means.

Disadvantages of K-Means++ Algorithm

- K-means++ clustering requires prior knowledge of K (or number of clusters) like K-Means.
- K-Means++ fails for non-linear data.

III. Overall System Architecture

A. System Details

The detail overall system architecture is shown below in the figure 1

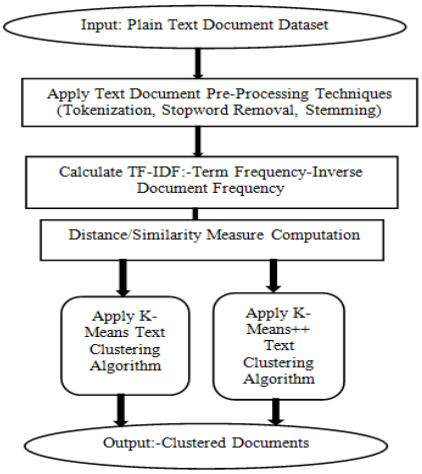


Figure. 1. Overall System Architecture

B. Document Illustration

Input to the Text Categorization System is document dataset containing 600 plain text documents of 3 categories – Festivals, Sports, and Tourism. Documents are collected from various news website.

- Festivals, 200 documents.
- Sports, 200 documents.
- Tourism, 200 documents.

C. Document Dataset Pre-processing Phase

This phase is used to remove words with less meaningful information. This pre-processing phase helps in reducing the clustering process time. Pre-processing of document includes three steps:

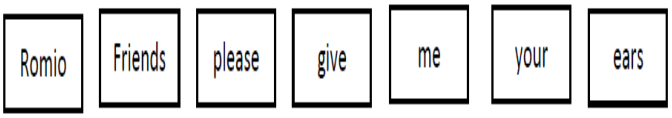
1)Tokenization: It is the method used for splitting the sentences into words by identifying the spaces, comma and special symbols among the words. So list of sentences and words are preserved for further processing.

Given a sequence of characters and document item, tokenization is basically the task of dividing it up into a pieces, called tokens.

For Example:

Input: Romio, Friends, please give me your ears.

Output:



2) Stopword Removal: Stop words are removed from text documents because in text mining applications these words are not considered as keywords. The most common Words found in in text documents are pro-nouns, prepositions, articles, etc. these words has no or very little information value in the text documents. These words are known as stop words. Stopword's Example: is, in, the, with, a, are, and, etc.

3) Stemming: This stemming technique identifies the stem/root form of the word. Stemming example: the words "cluster", "clustered", "clustering", "clusters" all these words can be rooted/stemmed to the word "cluster". The stemming algorithm used in our paper is Porter Stemmer algorithm which is the most popular stemming algorithm used in English language.

D. Tf-Idf Calculation

Once pre-processing of document dataset is done, each document in a dataset is represented as an N-dimensional vector d in term space, where "N" denotes total number of words/terms. Document vectors are exposed to some standard weighting schemes, such as Term Frequency-Inverse Document Frequency (TF-IDF), and standardized to have unit length.

We need to compute Term weights such as Term Frequency (TF), Inverse Document Frequency (IDF) and finally TF * IDF i.e. the product of TF and IDF.

TFIDF Analysis is done by taking into consideration two factors namely: Term Frequency (TF) and Inverse Document Frequency (IDF).

$$TF\text{-}IDF = TF * IDF \quad (1)$$

Where

TF: - K / W where K = total count of particular word appearing in a document d and T = total number of words in a document d.
IDF = D / DF where
D = number of documents in total in a document dataset,
DF = total number of documents containing a given word.

E. Similarity/Distance Measures

Before doing clustering of text documents, a distance/similarity measure need to be calculated. The measure helps in identifying the degree of similarity or difference between target documents. All text clustering algorithms have to adopt some cluster relationship among the documents. Similarity or dissimilarity between a pair of documents can be described either implicitly or explicitly. The nature of distance/similarity measure plays important role in the failure or success of a clustering technique. The description of some of the distance/similarity measures is given below in brief.

1) Cosine Similarity: Cosine Similarity is the most common similarity measure used in text document clustering.Main property of the Cosine Similarity measure is that it is independent of the document size.Consider 2 documents di and dj, the similarity between these documents can be computed as

$$cos(d_i, d_j) = \frac{d_i \cdot d_j}{\| d_i \| \| d_j \|} \quad (2)$$

Cosine values of the documents are restricted between [0, 1].When two documents has cosine value as 1 they are considered as similar, and if cosine value is 0 than documents are considered as dissimilar.

2) Jaccard Coefficient: The Jaccard coefficient also known as Tanimoto coefficient is a similarity measure.For text document,Jaccard Coefficient is defined as the intersection divided by the union of the document data objects. Jaccard coefficient matches the weighted sum of common terms to the weighted sum of terms that are present in any one of the two document but are not the common terms.

$$Sim_{eJacc}[u_i, u_j] = \frac{u_i^t u_j}{\|u_i\|^2 + \|u_j\|^2 - u_i^t u_j} \quad (3)$$

The value of the Jaccard coefficient is bounded between 0 and 1. Jaccard Coefficient value is 1 when the Ui = Uj and value is 0 when Uj and Ui are disjoint, where 1 means the two document data objects are similar and 0 means they are dissimilar.

3) Euclidean Distance: Euclidean Distance is considered as a fixed metric for statistical or geometrical problems. It is defined as common distance among two document data points and can be measured with a ruler in two-dimensional or three-dimensional space. For K-means algorithm, Euclidean Distance is used as a default distance measure. Euclidean distance is one of the most common distance measures:

$$Dist(di, dj) = \| |di - dj \| \quad (4)$$

Above formula of Euclidean Distance computation is used in the traditional k-means algorithm. In text documents, Euclidean Distance between two given documents da and db which are denoted by their term vectors ta and tb is measured as

$$D_E(\vec{t_a}, \vec{t_b}) = \left(\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2\right)^{1/2}, \quad (5)$$

Where T = {t1, . . . , tm} is the term set. Term weights are TF-IDF values.

IV. PERFORMANCE EVALUATION AND RESULTS

A. Experimental Setting Up

Text Categorization System is implemented in Java language using MARS java EE ide Version 4.5.2 on Windows 8 machine. For clustering 600 plain text documents of three categories: Festivals, Sports and Tourism;K-means and K-Means++ clustering algorithm have been used.

B. Performance Evaluation Measures

Measures used for evaluating the performance of the Text Categorization System are Purity and Entropy. Detail description of these performance measures is given below.

1)Purity: Purity measures cluster unity, that is degree to which a cluster holds documents from one single category. For a particular cluster j, purity of cluster j is calculated as

$$P_j = \frac{1}{n_j} \max_i(n_j^i), \quad (6)$$

Where max_i (n_j^i) is the number of documents in cluster j with category label i and n_j^i represents the number of documents from cluster j assigned to category i. In a simple way, purity of cluster j i.e. Pj is the part of the overall cluster size that is the largest categories of documents assigned to that cluster represents. The overall cluster purity is obtained as a weighted sum of each cluster purity. Overall purity of cluster is given as

$$Purity = \sum_{j=1}^{m} \frac{n_j}{n} P_j, \quad (7)$$

Where m represents number of clusters,n is total number of documents in dataset/corpus, nj represents total number of documents from dominent category. Larger purity value gives better clustering results.

2) *Entropy:* Measure cluster purity based on the given category label.Thus, if all the clusters contains documents from only a single category, the overall entropy is 0. But if,the category labels of documents in a cluster become more varied, the entropy value increases.

Given a set of clusters, we need to calculate entropy E of that cluster set. Firstly, we compute category distribution of

documents in each cluster, i.e., for individual cluster j we compute p_{ij}.Probability p_{ij} indicates that member of cluster j belongs to category i.Given this category distribution, the cluster j entropy (E) is computed as

$$E_j = -\sum_i p_{ij} log(p_{ij}) \quad (8)$$

Where the sum of all categories is taken. The overall entropy of cluster set is calculated as the weighted sum of all clusters entropies, as shown in below formula,

$$E = \sum_{j=1}^{m} \frac{n_j}{n} E_j \quad (9)$$

Where j is the individual cluster, nj is represented as size of cluster j, m indicates number of clusters, and n is the total number of documents in the cluster.

C. *Observations*

For both K-Means and K-Means++ clustering, the entropy and purity of the cluster have been computed for evaluation purpose. Our observation shows that K-Means++ Clustering Algorithm is better for clustering text documents as compared to K-means.

TABLE I. K-MEANS CLUSTERING ALGORITHM

Text Document Clustering Algorithm	**K-Means**		
Distance/Similarity Measures	**Cosine Similarity**	**Euclidean Distance**	**Jaccard Coefficient**
Purity	0.7050	0.6520	0.5883
Entropy	0.5625	0.5690	0.7705

TABLE II. K-MEANS++ CLUSTERING ALGORITHM

Text Document Clustering Algorithm	**K-Means++**		
Distance/Similarity Measures	**Cosine Similarity**	**Euclidean Distance**	**Jaccard Coefficient**
Purity	0.8216	0.6583	0.6466
Entropy	0.4176	0.5589	0.6768

TABLE III. COMPARSON of K-MEANS and K-MEANS++ CLUSTERING ALGORITHM

K-Means Based Text Clustering Algorithm	**K-Means++ Based Text Clustering Algorithm**
Partitioning Based Technique	Partitioning Based Technique
Input: document dataset, K number of clusters, K randomly chosen centroids, total number of iterations	Input: document dataset, K number of clusters, K randomly chosen first initial centroid, total number of iterations
Goal: Is to minimize sum of the squared distance	Goal: Is to minimize sum of squared distance
Time Complexity: K-Means algorithm takes lesser time to give output as compared to K-Means++	Time Complexity: K-Means++ algorithm takes more time to give output as compared to K-Means
Speed Complexity: K-Means algorithm execution speed is faster than K-Means++ algorithm	Speed Complexity: K-Means++ algorithm execution speed is slower than K-Means algorithm
Result: K-Means algorithm results are poor as compared to K-Means++	Result: K-Means++ algorithm results are much better as compared to K-Means
Exit condition: No changes in new cluster centroids.	Exit condition: No changes in new cluster centroids.

V. CONCLUSION

The K-Means is simple and popular clustering technique, but its results are based on choice of cluster centers so it easily results in local optimization. Since initial cluster centers are selected randomly in K-Means, they can be selected really badly. The K-means++ algorithm attempts to solve this problem, by evenly spreading the initial cluster centers. In K-means++ first initial cluster center is selected randomly and then it searches for other initial cluster centers given the first one based on definite probability. K-Means++ gives better results than K-Means.

Taking into consideration all Similarity/Distance measures, two algorithm comparison shows that the K-means++ using Cosine Similarity measure has highest value of purity and lowest value of entropy as compared to K-Means algorithm. K-Means and K-Means++ algorithm shows the best results with Cosine Similarity as compared to Jaccard Coefficient and Euclidean Distance. We also conclude that that the results of both the algorithms with Euclidean Distance are better than the results with Jaccard Coefficient.

REFERENCES

[1] Dawid Weiss, and Stanislaw Osi´nski, "A Concept Driven Algorithm for Clustering Search Results", 1541-1672/05/$20.00 © 2005 IEEE INTELLIGENT SYSTEMS.

[2] Saleh Alsaleem, Shaqra University, Saudi Arabia, "Automated Arabic Text Categorization Using SVM and NB", International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011.

[3] Sushma R. Vispute, M. A. Potey.2013.Automatic text categorization of marathi documents using clustering technique.IEEE transaction on Advanced Computing Technologies (ICACT), 2013 15th International Conference .

[4] David Arthur and Sergei Vassilvitskii." k-means++: The Advantages of Careful Seeding.

[5] Jaydeep Jalindar Patil, Nagaraju Bogiri..2015.Automatic Text Categorization Marathi Documents.International Journal of Advance Research in Computer Science and Management Studies Volume 3, Issue 3, March 2015.

[6] Pranjal Singh, Mohit Sharma, Prof. Amitabha Mukerjee.2013.Text Document Clustering and Similarity Measures.

[7] Ms.Anjali Ganesh Jivani.2011.A Comparative Study of Stemming Algorithms. Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938 NOV-DEC 2011.

[8] Parvesh Kumar,Siri Krishan Wasan.2010.Comparative Analysis of k-mean Based Algorithms. IJCSNS International Journal of Computer Science and Network Secu 314 rity, VOL.10 No.4, April 2010.

[9] Rakesh Chandra Balabantaray, Chandrali Sarma, Monica Jha.2013.Document Clustering using K-Means and K-Medoids.International Journal of Knowledge Based Computer System Volume 1 Issue 1 June 2013.

[10] Fabrizio Sebastiani, "A Tutorial on Automated Text Categorisation",Istituto di Elaborazione dell'Informazione Consiglio Nazionale delle Ricerche Via S. Maria, 46 - 56126 Pisa(Italy).