An Approach to Enhance the CPI Using Porter Stemming Algorithm

Dileep Reddy Chintala*
Scholar, Dept of Computer Science & Engineering,
Madanapalle Institute of Technology & Science,
Madanapalle-517325, India

E.Madhusudhana Reddy
Professor, Dept of Computer Science & Engineering,
Madanapalle Institute of Technology & Science,
Madanapalle-517325, India

Abstract— In document clustering the related documents are grouped into clusters. The Correlation Preserving Indexing (CPI) method is one of the methods used in document clustering. The CPI is the spectral clustering method and is performed in the correlation similarity measure space. In CPI method the documents are projected into a low-dimensional semantic space in which the correlations between the documents in the local patches are maximized while the correlations between the documents outside these patches are minimized simultaneously. For detecting the intrinsic geometrical structure of the document space the correlation as a similarity measure is more suitable than Euclidean distance because the intrinsic geometrical structure of the document space is often embedded in the similarities between the documents. However the speed and accuracy of the CPI method is not satisfactory. In this paper along with CPI, K-means clustering and Porter Stemming Algorithm are used to increase the speed and accuracy of the CPI method.

Keywords— document clustering, Euclidean distance, K-means clustering, porter stemming algorithm.

I. INTRODUCTION

Cluster analysis is an unsupervised learning method that constitutes a cornerstone of an intelligent data analysis process. The exploration of inter-relationships among a collection of patterns is done by cluster analysis by organizing them into homogeneous clusters. The cluster analysis is called unsupervised learning because unlike classification (known as supervised learning), no apriority labelling of some patterns is available to use in categorizing others and inferring the cluster structure of the whole data. The measure of the density of connections between the instances of a single cluster is called intra-connectivity. A high intra-connectivity indicates a good clustering arrangement because the instances grouped within the same cluster are highly dependent on each other. Inter-connectivity is a measure of the connectivity between distinct clusters. A low degree of interconnectivity is desirable because it indicates that individual clusters are largely independent of each other. Every instance in the data set is represented using the same set of attributes. The attributes are continuous, categorical or binary. To induce a hypothesis from a given data set, a learning system needs to make assumptions about the hypothesis to be learned. These assumptions are called biases. Since every learning algorithm uses some biases, it behaves well in some domains where its biases are appropriate while it performs poorly in other domains. A problem with the clustering methods is that the interpretation of the clusters may be difficult. In addition, even if there were no clusters in the data the algorithms will always assign the data to clusters. Therefore, it is essential to analyse whether the data set exhibits a clustering tendency if the goal is to make inferences about its cluster structure. Due to inaccurate measurement or due to missing values there may be errors (called noise) in the collected data set in a real-world application and therefore a pre-processing is needed (e.g. choose a strategy for handling missing attributes values). The choice of which specific learning algorithm to use is a critical step, too. The issue of relating the learning algorithms to the type of data and to the nature of the problem to be solved still remains an open and fundamental problem. An evaluation criterion of clustering quality is the unknown attribute prediction accuracy. The first step is by taking an unseen instance, removing the value of one of its attributes and then trying to classify it. The missing attribute on the unseen instance is predicted to be the same as the value of the attribute on the closest matching instance. This value can be then compared to the actual value of the removed attribute and so can be judged to be correct or not. For each attribute this process is repeated. The number of attributes correctly predicted is then totalled up and divided by the number of attributes in order to give the average prediction accuracy.

II. CLASSIFICATION WITH CLUSTERING

In this section, we give the intuition of the proposed algorithm in order to understand how clustering prepares the ground for classification. The algorithm consists of the following steps:

1. Clustering step: to cluster both the training and testing set.
2. Expansion step: to augment the dataset with meta-features originated from the clustering step.
3. Classification step: to train a classifier with the expanded dataset.
A classifier trained with the given training examples will probably find hyper plane A instead of the desirable hyper plane B, as shown in Fig. 1. In Fig. 2, both datasets (training and testing) are clustered into two non-overlapping clusters. In the ideal case, the two clusters contain the positive and negative examples of the whole data set respectively. Then, corresponding meta-features are propagated to the existing feature vectors, and all feature vectors inside the same cluster are augmented with the same meta-feature. The dataset is transformed into a new coordinate system. Since feature vectors inside the same cluster are augmented with the same attribute-value pair, these vectors are now closer to one another resulting to an increase in the dataset’s density, illustrated in Fig. 2. Increasing the inter-cluster distance consequently leads to the maximal margin hyper plane B, as shown in Fig. 3. In a way, the classifier is tuned to the testing set and the classification efficiency is expected to improve. Intuitively, the classifier with the largest margin will give lower expected risk, i.e. better generalization.

Fig. 1 Classification with the original data  
Fig. 2 Dataset after clustering

Fig. 3 Classification with the new expanded dataset.

A. Documentation Clustering

How the documents are organized for review can make a big difference in the efficiency of review, not only saving costs, but also improving accuracy by assigning similar documents to the same reviewer. The text in your documents is examined by clustering software and the related documents are determined and group them into clusters. Clustering makes it easy to explore and categorize "big data" sets of documents, bringing efficiency to the electronic discovery process. Clustering does the electronic equivalent of putting your documents into labelled boxes so that things only end up in the same box if they belong together. This allows you to explore and manage your documents by browsing through a relatively small set of boxes (clusters) instead of digging through the much bigger data set of documents directly. It organizes the documents according to the structure that arises naturally, without preconceptions or query terms. To provide a quick overview of the cluster it labels each cluster with a set of keywords. It also identifies a "representative document" that can be used as a proxy for the cluster.
B. K-Means Clustering

In statistics and data mining, **K-means clustering** is a method of cluster analysis which aims to partition \( n \) observations into \( K \) clusters in which each observation belongs to the cluster with the nearest mean. This results into a partitioning of the data space into Voronoi cells.

The problem is computationally difficult (NP-hard), however there are efficient heuristic algorithms that are commonly employed that converge fast to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, to model the data they both use cluster centres, however K-means clustering tends to find clusters of comparable spatial extend, while the expectation-maximization mechanism allows clusters to have different shapes.

C. The K-Means Clustering Algorithm

- \( X_1, \ldots, X_n \) are data points or vectors or observations
- Each observation will be assigned to one and only one cluster
- \( C(i) \) denotes cluster number for the \( i \)th observation
- Dissimilarity measure: Euclidean distance metric
- \( K \)-means minimizes within-cluster point scatter:

\[
W(C) = \frac{1}{2} \sum_{i=1}^{K} \sum_{C_{ij} \neq 0} \| x_i - x_j \|^2 = \sum_{i=1}^{K} N_i \sum_{j=1}^{C_{ij}} \| x_i - m_j \|^2
\]

where

- \( m_j \) is the mean vector of the \( j \)th cluster
- \( N_i \) is the number of observations in \( i \)th cluster

(a) K-means clustering, for \( K = 3 \). The input data (black circles) are assigned to final clusters after two iterations. (b) Agglomerative clustering. Two-dimensional data points A–F from the diagram shown on the right, which, when partitioned along the dashed horizontal line, results in three clusters as indicated by the circled data points. (c) Normalized Cuts. The blue line partitions the graph into \( A \) (black nodes) and \( B \) (red nodes). Here, \( \text{cut}(A, B) \) is the sum of the weights of the dashed (removed) edges, \( \text{assoc}(A, V) \) is the sum of the weights of the black edges and dashed edges, and \( \text{assoc}(B, V) \) is the sum of the weights of the red edges and the dashed edges. (For interpretation of the references to colour in this figure legend, the web version of this article is referred by the reader.)

D. Applications of the Algorithm

The K-means clustering in particular when using heuristics such as Lloyd’s algorithm is rather easy to implement and apply even on large data sets. As such, in various topics, ranging from market segmentation, computer vision, geostatistics and astronomy to agriculture it has been successfully used. For other algorithms it is often used as a pre-processing step, for example to find a starting configuration.
E. Mean Shift Clustering

Basic mean shift clustering algorithms maintain a set of data points the same size as the input data set. Initially, this set is copied from the input set. Then the mean of those points in the set that are within a given distance of that point replaces this set. By contrast, this updated set is restricted by $K$-means to $K$ points usually much less than the number of points in the input data set, and replaces each point in this set by the mean of all points in the input set that are closer to that point than any other (e.g. within the Voronoi partition of each updating point). The likelihood mean shift is the mean shift algorithm that is similar then to $K$-means and replaces the set of points undergoing replacement by the mean of all points in the input set that are within a given distance of the changing set\(^{[11]}\). One of the advantages of mean shift over $K$-means is that there is no need to choose the number of clusters, because mean shift is likely to find only a few clusters if indeed only a small number exist. However, $K$-means can be much faster than mean shift. Mean shift has soft variants much as $K$-means does.

III. Porter Stemmer

The Porter stemming algorithm is a common stemming algorithm which works well for English. It reduces words such as "stem", "stems", "stemming" to a single root, e.g., "stem". The root is not always a real English word. Therefore, after the stemming of all the words, to obtain a more readable WordCloud, each stemmed-word is replaced by the shortest word of its family. It should be noted that some words are removed from the text before finding the keywords. First, remove all common English words such as: the, is, or are. In the following file the list of these words can be found. Then common biological terms such as: contribute, experiments, abstracts are removed. Here these terms are available. After experimenting with many WordClouds these terms were chosen by hand curation. Removed common terms are also contained in gene ontology annotations and text-mining of generifs. Finally, other terms such as the input gene names, the author name, or the keywords of the Pub med search are also removed to avoid self-referencing. The Porter stemming algorithm (or ‘Porter stemmer’) is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems. In linguistic morphology and information retrieval, stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish". On the other hand, "argued", "argue", "argues", "arguing", and "argus" reduce to the stem "argu" (illustrating the case where the stem is not itself a word or root) but "argument" and "arguments" reduce to the stem "argument". Suffix stripping algorithms may differ in results for a variety of reasons. One such reason is that the output word which the algorithm constrains be a real word in the given language or not. Some approaches do not require the word to actually exist in the language lexicon (the set of all words in the language). Alternatively, some suffix stripping approaches maintain a database (a large list) of all known morphological word roots that exist as real words. The list is checked by these approaches for the existence of the term prior to making a decision. Typically, alternate action is taken, if the term does not exist. Several other criteria might be involved in this alternate action. The non-existence of an output term may serve to cause the algorithm to try alternate suffix stripping rules.
IV. Clustering Results

A. 20 Newsgroups (or NG20)

On the 20 newsgroups (or NG20) data set experiments were performed. Under the same experimental setting the proposed algorithm is compared with the other competing algorithms. When the number of nearest neighbours is set to seven or eight, the experimental results of K-means on NG20 data set are obtained. Our algorithm performs better than or competitively with other algorithms in all experiments. The experiments details can be described as follows.

Table I

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Accuracy (%)</th>
<th>Pearson-Kmeans &lt; p-Kmeans &lt; p-QR &lt; Spectral &lt; LPI &lt; CPI &lt; Enhanced CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NG1/NG2</td>
<td>74.14±16.10</td>
<td>78.99±15.84</td>
</tr>
<tr>
<td>NG2/NG3</td>
<td>63.92±11.49</td>
<td>68.24±12.54</td>
</tr>
<tr>
<td>NG8/NG9</td>
<td>67.78±14.18</td>
<td>71.28±13.29</td>
</tr>
<tr>
<td>NG10/NG11</td>
<td>64.50±10.87</td>
<td>64.50±10.54</td>
</tr>
<tr>
<td>NG11/NG15</td>
<td>73.56±13.51</td>
<td>74.02±8.68</td>
</tr>
<tr>
<td>NG18/NG19</td>
<td>65.38±6.75</td>
<td>60.08±10.97</td>
</tr>
</tbody>
</table>

Note that "<<" (">>") indicates that schemes on the right are significantly better (worse) than the schemes on the left, and "<" (">")) indicates that the relationship is not significant. In all results of statistical significance tests, the expression $A < B < C$ means the relationship between A and C is $A < C$.

Roughly 20,000 documents that come from 20 specific Usenet newsgroups are present in the 20 newsgroups corpus. To illustrate the performance of the proposed CPI algorithm and other competing algorithms we repeated the experiments in [17] and [26]. Binary clustering is involved in the first set of experiments. In each experiment, to obtain statistically reliable clustering result we randomly chose 50 documents from the two selected newsgroups and 100 runs were conducted for each algorithm. In Table the means and standard deviations of the test results were recorded. Under same experimental setting, we also tested other competing methods including K-means, p-Kmeans [17], p-QR [17], and Spectral [26]. It can be seen from Table on all six data sets the CPI achieves the best clustering accuracy. The second best is performed by LPI, the worst is performed by K-means and p-Kmeans and p-QR outperform K-means. CPI also performs the best under normalized mutual information metric. The p-QR or p-Kmeans performs the worst and the K-means performs better than p-QR and p-Kmeans. Using paired t-test with a significance level of 0.05 we also tested the significance of the comparisons between various methods. We can see that in most of the datasets the CPI method.
Chintala et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(7), July - 2013, pp. 1148-1156

outperforms with statistical significance other competing methods. The c-way clustering with c=5 and c=10 is involved in the second set of experiments. Again for each test 100 runs were used and in Table 2 the means and standard deviations were recorded.

### TABLE II

**COMPARISON OF PERFORMANCE OF DIFFERENT CLUSTERING METHODS USING NG20 DATA CORPUS**

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Accuracy (%)</th>
<th>K-means</th>
<th>p-Kmeans</th>
<th>p-QR</th>
<th>Spectral</th>
<th>LPI</th>
<th>CPI</th>
<th>Enhanced CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NG2/NG3/NG4/NG5/NG6 (50)</td>
<td>44.66±7.47</td>
<td>45.20±7.22</td>
<td>45.8±6.08</td>
<td>44.10±5.56</td>
<td>54.40±7.18</td>
<td>58.06±6.18</td>
<td>63.85±7.96</td>
<td></td>
</tr>
<tr>
<td>NG2/NG3/NG4/NG5/NG6 (100)</td>
<td>41.21±6.55</td>
<td>42.79±6.38</td>
<td>42.53±4.43</td>
<td>45.08±4.89</td>
<td>55.40±5.97</td>
<td>55.00±2.18</td>
<td>54.42±2.03</td>
<td></td>
</tr>
<tr>
<td>NG2/NG9/NG10/NG15/NG18 (50)</td>
<td>55.24±9.18</td>
<td>56.76±10.25</td>
<td>58.12±9.01</td>
<td>66.48±9.29</td>
<td>80.52±11.32</td>
<td>84.84±10.30</td>
<td>88.73±12.68</td>
<td></td>
</tr>
<tr>
<td>NG2/NG9/NG10/NG15/NG18 (100)</td>
<td>50.29±9.61</td>
<td>54.20±9.20</td>
<td>55.62±8.56</td>
<td>70.68±8.96</td>
<td>82.46±12.01</td>
<td>87.52±10.13</td>
<td>90.47±15.31</td>
<td></td>
</tr>
<tr>
<td>NG1/NG5/NG7/NG8/NG11/NG12/NG13/NG14/NG15/NG17 (50)</td>
<td>45.68±6.68</td>
<td>46.62±6.05</td>
<td>47.89±5.55</td>
<td>52.89±5.35</td>
<td>60.32±7.72</td>
<td>61.56±6.96</td>
<td>65.27±8.91</td>
<td></td>
</tr>
<tr>
<td>NG1/NG5/NG7/NG8/NG11/NG12/NG13/NG14/NG15/NG17 (100)</td>
<td>42.06±6.25</td>
<td>42.64±6.07</td>
<td>44.38±5.12</td>
<td>54.36±4.86</td>
<td>61.63±6.82</td>
<td>67.81±6.79</td>
<td>71.41±6.93</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Statistical significance test based on accuracy values (p-value = .05)

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Normalized mutual information (%)</th>
<th>K-means</th>
<th>p-Kmeans</th>
<th>p-QR</th>
<th>Spectral</th>
<th>LPI</th>
<th>CPI</th>
<th>Enhanced CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NG2/NG3/NG4/NG5/NG6 (50)</td>
<td>27.43±9.39</td>
<td>28.01±8.57</td>
<td>28.04±8.56</td>
<td>32.30±7.12</td>
<td>38.05±7.54</td>
<td>43.28±8.26</td>
<td>47.35±10.32</td>
<td></td>
</tr>
<tr>
<td>NG2/NG3/NG4/NG5/NG6 (100)</td>
<td>21.78±7.52</td>
<td>21.06±9.18</td>
<td>21.45±8.54</td>
<td>29.25±5.42</td>
<td>36.25±7.51</td>
<td>43.35±6.79</td>
<td>49.03±9.83</td>
<td></td>
</tr>
<tr>
<td>NG2/NO9/NG10/NG15/NG18 (50)</td>
<td>40.54±10.91</td>
<td>39.83±10.68</td>
<td>40.26±9.45</td>
<td>52.62±8.42</td>
<td>72.25±8.47</td>
<td>74.12±9.88</td>
<td>79.21±11.24</td>
<td></td>
</tr>
<tr>
<td>NG2/NO9/NG10/NG15/NG18 (100)</td>
<td>32.63±11.15</td>
<td>36.82±9.65</td>
<td>37.06±9.01</td>
<td>61.37±10.51</td>
<td>74.71±11.44</td>
<td>84.21±10.68</td>
<td>88.38±12.03</td>
<td></td>
</tr>
<tr>
<td>NG1/NG5/NG7/NG8/NG11/NG12/NG13/NG14/NG15/NG17 (50)</td>
<td>50.58±6.07</td>
<td>50.80±5.79</td>
<td>51.11±5.23</td>
<td>55.34±5.16</td>
<td>62.94±6.81</td>
<td>64.18±6.02</td>
<td>69.73±6.94</td>
<td></td>
</tr>
<tr>
<td>NG1/NG5/NG7/NG8/NG11/NG12/NG13/NG14/NG15/NG17 (100)</td>
<td>39.58±5.86</td>
<td>42.87±6.81</td>
<td>43.10±6.81</td>
<td>56.10±5.56</td>
<td>63.15±5.70</td>
<td>66.81±4.65</td>
<td>70.71±7.03</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Statistical significance test based on normalized mutual information (p-value = .05)

The number of random documents chosen from the newsgroups sets is indicated by the number in the parenthesis (50 or 100). As can be seen, in all six data sets CPI achieves the best accuracy and normalized mutual information. Under accuracy metric, p-Kmeans and p-QR perform better than K-means. The K-means outperforms the p-QR and p-K-means methods in two data sets under normalized mutual information metric. The statistical significance test results show that CPI is more accurate than the other methods with statistical significance for most of the data sets.

### B. Term Frequency

Each document is represented as a term frequency vector. The term frequency vector can be computed as follows:

1. Transform the documents to a list of terms after words stemming operations.
2. Remove stop words. Stop words are common words that contain no semantic content.
3. Compute the term frequency vector using the TF/IDF weighting scheme.

**TF/IDF** is the product of two statistics called term frequency and inverse document frequency. For determining the exact values of both statistics various ways are existed. In the case of the **term frequency** \( tf(t,d) \), the simplest choice is to use the **raw frequency** of a term in a document, i.e. the number of times that term \( t \) occurs in document \( d \). If we denote the raw frequency of \( t \) in \( d \) by \( f(t,d) \), then the simple tf scheme is \( tf(t,d) = f(t,d) \). Other possibilities include:

- **boolean “frequencies”**: \( tf(t,d) = 1 \) if \( t \) occurs in \( d \) and 0 otherwise;
- **logarithmically scaled frequency**: \( tf(t,d) = \log (f(t,d)+1) \);
- **augmented frequency**, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document:

\[
    tf(t,d) = 0.5 + \frac{0.5 \times f(t,d)}{\max\{f(w,d) : w \in d\}}
\]
The measure of whether the term is common or rare across all documents is called the inverse document frequency. By dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient, the inverse document frequency is obtained.

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

with

- $|D|$: cardinality of $D$, or the total number of documents in the corpus
- $|\{d \in D : t \in d\}|$: number of documents where the term $t$ appears (i.e., $t_f(t, d) \neq 0$). This will lead to a division-by-zero if the term is not in the corpus. Therefore it is common to adjust the formula to

$$1 + |\{d \in D : t \in d\}|$$

Mathematically the base of the log function does not matter and a constant multiplicative factor towards the overall result is constituted. Then tf–idf is calculated as

$$t_f\text{idf}(t, d, D) = t_f(t, d) \times \text{idf}(t, D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf’s log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0. Various (mathematical) forms of the tf–idf term weight can be derived from a probabilistic retrieval model that mimics human relevance decision making.

The term frequency vector can be computed as follows:
1. Transform the documents to a list of terms after words stemming operations.
2. Remove stop words. Stop words are common words that contain no semantic content.
3. Compute the term frequency vector using the TF/IDF weighting scheme.

### TABLE III

<table>
<thead>
<tr>
<th>Document set</th>
<th>ADDC</th>
<th>$K$-means</th>
<th>HKA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Euclidian</td>
<td>Cosine</td>
<td>Euclidian</td>
</tr>
<tr>
<td>DS1</td>
<td>0.7184</td>
<td>0.7849</td>
<td>0.4638</td>
</tr>
<tr>
<td>DS2</td>
<td>0.6415</td>
<td>0.7032</td>
<td>0.4517</td>
</tr>
<tr>
<td>DS3</td>
<td>0.7425</td>
<td>0.7925</td>
<td>0.5287</td>
</tr>
<tr>
<td>DS4</td>
<td>0.4153</td>
<td>0.4419</td>
<td>0.2510</td>
</tr>
<tr>
<td>DS5</td>
<td>0.8425</td>
<td>0.9236</td>
<td>0.6170</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Document set</th>
<th>F-measure $K$-means</th>
<th>HKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>0.5632</td>
<td>0.7662</td>
</tr>
<tr>
<td>DS2</td>
<td>0.5202</td>
<td>0.7824</td>
</tr>
<tr>
<td>DS3</td>
<td>0.6117</td>
<td>0.8968</td>
</tr>
<tr>
<td>DS4</td>
<td>0.7236</td>
<td>0.8692</td>
</tr>
<tr>
<td>DS5</td>
<td>0.4236</td>
<td>0.6805</td>
</tr>
</tbody>
</table>

C. Experimental Results
D. Comparison with Other Algorithms

To demonstrate how our method improves the document clustering accuracy in comparison to the best contemporary methods, we implemented three known partitioning algorithms namely Genetic K-means (GA), Particle Swarm Optimization based clustering (PSO) and a Mises-Fisher Generative Model based algorithm (GM).

E. Runtime Analysis

For the document numbers ranging from 1000 to approximately 10,000, the evaluations were conducted. 10 test runs were conducted on different randomly chosen documents, for each given document number and the final performance scores were obtained by averaging the scores from the all tests.

V. CONCLUSION

In this Paper we present a new document clustering method based on K-means clustering method using Porter Stemming process, classification, CPI implementation and frequency calculation operation. Extensive experiments on NG20, corpora show that the proposed System K-means clustering method outperforms other classical clustering methods. Furthermore, the K-means method has good generalization capability and thus it can effectively deal with data sizes. CPI technique and Bayes algorithm are used in this project and also using the frequency like term frequency CTF, DF, IDF. Furthermore, the CPI method has good generalization capability and thus it can effectively deal with data with very large size.
REFERENCES


