Document Clustering Based On Semi-Supervised Term Clustering

Hamid Mahmoodi[1] and Eghbal Mansoori[2]

[1]Department of computer science and engineering, Shiraz University, Shiraz, Iran
hmahmoodi@cse.shirazu.ac.ir
[2]Department of computer science and engineering, Shiraz University, Shiraz, Iran
mansoori@shirazu.ac.ir

ABSTRACT

The study is conducted to propose a multi-step feature (term) selection process and in semi-supervised fashion, provide initial centers for term clusters. Then utilize the fuzzy c-means (FCM) clustering algorithm for clustering terms. Finally assign each of documents to closest associated term clusters. While most text clustering algorithms directly use documents for clustering, we propose to first group the terms using FCM algorithm and then cluster documents based on terms clusters. We evaluate effectiveness of our technique on several standard text collections and compare our results with the some classical text clustering algorithms.

KEYWORDS

Term Clustering, Fuzzy C-Means algorithm, Semi-supervised Feature Selection, Document Clustering

1. INTRODUCTION

Despite of using powerful feature selection methods, one of the main problems in document clustering domain is effect of many noisy, unrelated, short and uninformative documents in text datasets. Term clustering is directed to cluster a small and perfect set of terms in order to avoid such a noisy sample space.

Among the various text clustering domain methods, term clustering has been motivated more in language modeling (LM) areas and information retrieval systems which tend to extend sentences and queries with similar and close terms to improve sentence retrieval performance, especially in Question Answering systems. Other motivations except query expansion [1] for term clustering, can point to machine translation [2], text categorization [3], speech recognition [4], automatic spelling correction [5] and automatic thesaurus generation [6]. First time, brown et al. (1992) proposed a term clustering algorithm base on Average Mutual Information (AMI) between adjacent clusters [7]. First initialized clusters by single term and in a bottom-up approach combined cluster pairs which offer minimum decrease in their AMI. Momtazi introduced a class-based LM approach using term clustering in sentence retrieval for QA systems to solve data sparsity and exact matching problems [8]. Dagan et al. (1999) used word similarity to assign probabilities to unseen bigram to yield up to 20% perplexity improvement in prediction of unseen bigrams [9]. Pereira et al. (1993) suggested a fuzzy distributional word clustering schema, such that membership of each word in each categories was probabilistic. Based on co-occurrences between words, they modeled probabilities of words by averaged co-occurrence probabilities of word clusters [10]. Slonim et al. (2001) introduced an agglomerative word clustering approach

DOI : 10.5121/ijaia.2012.3306

and used these clusters as features for supervised document classification [11]. The idea was based on an information theoretic framework, which was termed as information bottleneck (IB) [12] method and was used to find word clusters. The IB method measures distortion between two joint distribution of two random variables X and Y. One variable is compacted, while preserves maximum mutual information over the other variable. One variable stood for word clusters W and other variable corresponded to document categories C. In agglomerative approach, |W| words was partitioned to singleton clusters and then iteratively, merged two word clusters into a new cluster in a way that minimized loss of mutual information over the categories. Mutual Information distributions (i.e. prior probability of word clusters, membership probabilities and distribution over the relevance variable) calculated with Jensen-Shannon (JS) divergence function. Finally, measured the probability of documents using bayes rule and chainlike related to word clusters. Massih et al. (2007) utilized word clusters to expand the question and title keywords to build an extractive summarizer system [13]. They first produced a small set of candidate sentences (1/4 of all sentences). Then, scored each sentences based on combination of some heuristic features from clusters of terms. They perform term clustering by unsupervised learning algorithm that was a classification variant of the well known Expectation Maximization algorithm. A summary produced by selecting the 10 highest scored sentences and constructed the final summary with at most 250 words in post processing step.

In most of term clustering studies, the notions of co-occurrence between terms in same document, paragraph, passage, sentences or a fixed-size window has been used for test word associations.
In this paper we are going to propose a multi-step feature selection process that aim to provide the most discriminative terms in the vocabulary, which usually encompass less than five percent of vocabulary size. We consider terms as bag-of-documents and represent each term as vector of documents, indexed by tf-idf [14] weighting scheme. We used the selected terms and incorporated limited labeled documents provided by the supervision, and initialized term clusters centers by taking average between terms vectors which were in joint between selected terms and most corresponding cluster's labeled documents. Most of the real-life text datasets have classes which are somewhat close to each other. For example consider Datasets re0 and re1 which are subsets of Reuters21587 text collection. We can't assert that some terms like "money", "bank" or "loan" belong to special classes like money, trade, interest or reserves. Therefore we believe that any term belongs to every class with the degree of membership. Consequently we decided to use fuzzy c-means clustering algorithm for clustering the terms. K-Means [15] algorithm and its extensions like c-means, spherical K-Means [16] and EM algorithm start with the random initialization and usually causes to unreliable and fluctuation results. We utilized few information provided by an expert and in a semi-supervised fashion, seeded clusters centers to solve this problem. After convergence of clustering algorithm, assigned each document of corpus to closest associate term cluster. The rest of this paper is organized as follows. Section 2 describes preprocessing steps applied on documents. Our proposed multi-step feature selection and term clusters centers initialization process is presented in section 3. Section 4 introduce fuzzy c-means clustering algorithm. Section 5 reports a set of experimental results on two skew subsets of reuters-21587 corpus and two text datasets containing of newsgroup messages.

2. PREPROCESSING

Here we consider documents to be bags of terms and denote $V = \{w_j\}, j = 1, \dots, |V|$ the set of V vocabulary terms and $D = \{d_i\},\ i = 1, \dots, N$ the set of N documents exist in the corpus. After tokenization, removing of stop words, numbers and mixed alphanumeric strings and elimination of rare words (occurring in less than 4 documents) and lowering uppercases characters, stemmed reminded terms using porter stemmer algorithm introduced in [17]. Then constructed the term-document matrix such that each entry of matrix indicated the frequency with which a

corresponding row term appears in corresponding column document. Then all entries of matrix transformed using *tf-idf* weighting scheme as follow:

$$f(t, d_i) = tf(t, d_i) log\left(\frac{N}{df(t, d_i)}\right), i = 1, \dots, N \quad (1)$$

Where *tf (t, d_i)* is frequency of term *t* appeared in d_i and *df (t, d_i)* is number of documents within the corpus in which the term occurs in. Finally normalized each column (document) by its length to have unit Euclidian length.

3. FEATURE SELECTION

Feature selection has been shown that is an inseparable part of many machine learning application, particularly in text mining scope which almost comes from distributions with the thousands of features. Because of time and memory limitations and existence of many noisy and irrelevant features which causes to high degrade in performance and precision of algorithms, feature selection greatly solve these problems by selecting best portion of features.

3.1. Common Unsupervised Feature Selection Methods

Generally all proposed feature selection techniques work in two main approaches which briefly introduce them in the following:

3.1.1. Filter Methods

Filter methods evaluate each feature, independent of special learning algorithm and give a score for each feature based on its fitness on a predefined objective function and rank it based on its score. Some of the most important known unsupervised filter based feature selections in text mining is Document Frequency (DF) [18] which assess each term based on the number of documents within the corpus in which the term occur in, Term Strength (TS) which measure the probability that a feature occurs in the second half of a pair of related documents condition on that it appeared earlier[19], Entropy-based feature ranking[20] which measure score of each term based on its entropy reduction when it is removed, Term Contribution (TC) [21] which is extension of DF and measure rank of terms based on their overall contribution on entire documents similarity. Information Gain [22] and χ statistic (CHI) [23] are the most known supervised filter methods.

3.1.2. Wrapper Methods

Wrapper approaches evaluate a feature subset according to the performance of the unsupervised learning algorithm on the original data projected onto the features in the subset [24]. Wrapper methods employ some search strategies such as greedy hill climbing or simulated annealing search techniques to find the best subset of features in search space and fall in brute-force methods that need great deal of computation to cover all search space, but despite of high computation complexity, wrappers have been shown robust against overfitting [22]. Two common greedy searches which usually incorporate in wrappers are forward subset selection which progressively incorporate most promising features into larger subset and backward subset selection which start with the set of all features and progressively eliminate least promising ones [26].

3.2. Proposed Feature Selection Process

Here, we propose two consecutive step feature selection to construct few most important features. In the first step we use Term Contribution (TC) feature selection method which is a known and powerful filter method. We selected the 50 percent of higher scored features and used them in the

second step. In the second step, we proposed simple and new feature selection method based on maximum index value (*tf-idf*) of each document.

3.2.1. Term Contribution (TC)

TC proposed in [21] and we used this filter approach feature selection measure for the first phase of our feature selection process and selected 50 percent of vocabulary terms which had higher TC score. TC is special extension of DF which is based on documents similarity. DF gives equal importance for each term in different documents, and causes to bias by those terms which have high document frequency but uniform distribution over different classes [21]. This problem is more visible in skew datasets which distribution of classes is very different. Similarity between two documents d_i and d_j is defined as dot product:

$$sim(d_i, d_j) = \sum_t f(t, d_i) f(t, d_j), \quad i, j = 1, \dots, N \tag{2}$$

Where *f(t,d)* represents the *tf-idf* weight of term *t* in document *d*. Contribution of a term is defined as its overall contribution on entire documents similarity and is defined as:

$$TC(t) = \sum_{i \neq j} f(t, d_i) f(t, d_j), \quad i, j = 1, \dots, N \tag{3}$$

The time complexity of TC is of order $O(M\overline{N^2})$ where M is the number of all terms and $\overline{N}$ is the average documents number per term occurs in. We preferred TC to other unsupervised term scoring methods for two reasons. First its better time complexity relative to EN and ST due to small amount of $\overline{N}$ for most terms and second, unlike DF which biased on terms with high document frequency, take into account the importance of terms in different documents.

We calculated the TC of all terms and sort them based on the scores which TC gave for each one. We observed in all our experiments less than 40 percent of higher ranked terms are worthwhile, but for confidence we chose at most 50 percent of terms which had higher TC score.

3.2.2. Most Frequent Best Terms (MFBT)

In the second phase of our feature selection process, we introduced a new simple measure to pickup most salient and important terms among selected terms in the previous phase. Formally, all terms which had highest and next-highest *tf-idf* weight in most documents were selected. Here *MFBT (t)* is the number of documents which term *t* has max and next-max weight within them.

$$MFBT = \left| t = w_k : k = \operatorname{argmax}_{w_j \in d_i} f(w_j) \;\; OR \;\; t = w_{k'} : k' = \operatorname{argmax}_{\substack{w_j \in d_i \\ j \neq k}} f(w_j) \right|_{d_i \in D} \tag{4}$$

Where |.| enumerates the number of times which *t* satisfies one of two conditions in (4) for all documents in the corpus. *D* is documents set in the corpus, *f (w)* is entry value (*tf-idf*) of term-document matrix for term *w* in a document. The experiments on various datasets showed that often most important terms in each document occurs in the first and second highest *tf-idf* weight of terms. Based on kind of distribution of classes in datasets, we choose those terms which their MFBT measure is more than a threshold and call them as final terms. From now on, we use *w* in equations, subscripts and explanations to denote the terms exist in final terms. Note that previous phase is essential to result of this phase, because there are many terms which their frequency are high for the sake of appear repeatedly in some docs and leads to have a high MFBT score, while they have low TC score for the sake of having low contribution in Docs similarity.

3.2.3. Semi-Supervised initialization of term cluster centers

In the final phase of our feature selection process, we initialized predefined number of term clusters centers with taking average between few selected terms vectors for each term clusters among final terms. For this, we took advantage of the limited labeled documents provided by an expert. Here, we supposed expert had enough skill to provide few best labeled documents for each topic (cluster). Selected terms for centers calculation had been chosen such that either exist in the final terms and either occur in the most corresponding cluster's labeled documents. Whatever selected term occur in more provided label document, it is more probable sufficient for that cluster center. The underlying assumption is that terms with the close semantic relation, more tend to appear in the same documents.

4. FUZZY CLUSTERING ALGORITHM

In this section we review one of the fuzzy clustering algorithms which is fully described in [25]. We more focus on Fuzzy c-means clustering algorithm and ignore description for other variant of clustering algorithms as they aren't appropriate for our needs and just use them for comparison needs. The goal of all clustering algorithms is based on min/maximization of their associated objective function. An objective function is a mathematical criterion which measure the quality of cluster models. We use following syntactic definitions in algorithms, equations and explanations.

$X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_w}\}$ Set of *w* terms object vector

$C = \{\overrightarrow{c_1}, \dots, \overrightarrow{c_k}\}$ Set of *k* clusters centers

$d(\overrightarrow{c_j}, \overrightarrow{x_i})$ Dissimilarity between center *j* and object *i*

$\overrightarrow{u_j} = \{u_{1j}, \dots, u_{kj}\}$ Membership vector of term object *j*

$U = \{\overrightarrow{u_1}, \dots, \overrightarrow{u_w}\}$ Membership matrix of size k×w

4.1. Fuzzy c-means Algorithm (FCM)

Unlike the hard clustering algorithm which required to each object belongs to exactly one cluster (i.e. $u_{ij}=\{0,1\}$), fuzzy clustering relax this requirement to $u_{ij}=[0,1]$ and constraint holds:

$$\sum_{i}^{k} u_{ij} = 1, \forall\, j \in \{1, \dots, w\}.$$

FCM is to minimize the objective function:

$$J_{FCM=\sum_{i=1}^{k}\sum_{j=1}^{w} u_{ij}^{m} d_{ij}^{2}} \quad (5)$$

Parameter *m*, $m>1$, is called fuzziness level. In [26] has been Shown that for $m=1$, FCM becomes identical to Hard c-means (K-Means). More value of *m* causes to increase in fuzziness of FCM. Membership value of object *j* to cluster *i* calculate as follows:

$$u_{ij} = \left[\frac{d(x_j,c_i)^{\frac{1}{m-1}}}{\sum_{l=1}^{k} d(x_j,c_l)^{\frac{1}{m-1}}} \right]^{-1} \quad (6)$$

Equation for recalculating clusters centers is:

$$\overrightarrow{c_i} = \frac{\sum_{j=1}^{w} u_{ij}^{m} \overrightarrow{x_j}}{\sum_{j=1}^{w} u_{ij}^{m}} \quad (7)$$

Note that in (6), not only u_{ij} depends on dissimilarity of object x_j and center c_i , but also depends on distances of object x_j to other clusters centers.

Several measures can be used for dissimilarity measures between objects and centers. The most frequently measure is the *Lp* norm with *p=2* as Euclidian distance:

$$d(x_j, c_i) = \|x_j - c_i\| = \sqrt{\sum_{n=1}^{s} (x_{jn} - c_{in})^2} \quad (8)$$

Where *s* is the dimensionality of the vectors. Due to effect of document length on Euclidean distance, we used cosine based dissimilarity measure which takes into account the angle between documents as follows:

$$d(x_j, c_i) = e^{-sim(x_j,c_i)} \quad (9)$$

Where $sim(x_j, c_i)$ is defined as:

$$sim(x_j, c_i) = \frac{\sum_{n=1}^{s} x_{jn} c_{in}}{\sqrt{\sum_{n=1}^{s} x_{jn}^2 \sum_{m=1}^{s} c_{in}^2}} \quad (10)$$

Steps of FCM are described briefly in follow:
Step1: Given *w* data points $X = \{\overrightarrow{x_1}, \dots, \overrightarrow{x_w}\}$, fix the number of centers *k*, $2 \leq k < w$, determine value of *m*, random initialization of membership matrix *U*.

Step2: Set p=0,1,2,…, compute k cluster center with equation (7)

Step3: Update U^p to U^{p+1} with equation (6)

Step4: Stop if $\|U^p - U^{p-1}\| < \epsilon$ or reach the predefined number of iterations else *p=p+1* and $U^p = U^{p+1}$.Go to stop 2.

Here, We instead of random initialization of membership matrix in step1 first initialized *k* center vectors which completely described in section 3 and replaced order of step3 and step2 in FCM algorithm.

4.2. Document Clustering

Finally, after convergence of term clustering algorithm, we assigned each document to its closest associate term cluster as follows:

$$d_i \in c_j \; if \;\; j = argmax_p \sum_{l=1}^{w} f(t_l, d_i) u_{pl} \quad (11)$$

Where u_{pl} is the membership value of term t_l in term cluster c_p and c_j is the j'th associated term cluster and $f(t_l, d_i)$ is term-document matrix value of term l in document i.

The outline of entire our proposed method is depicted in Figure 1.

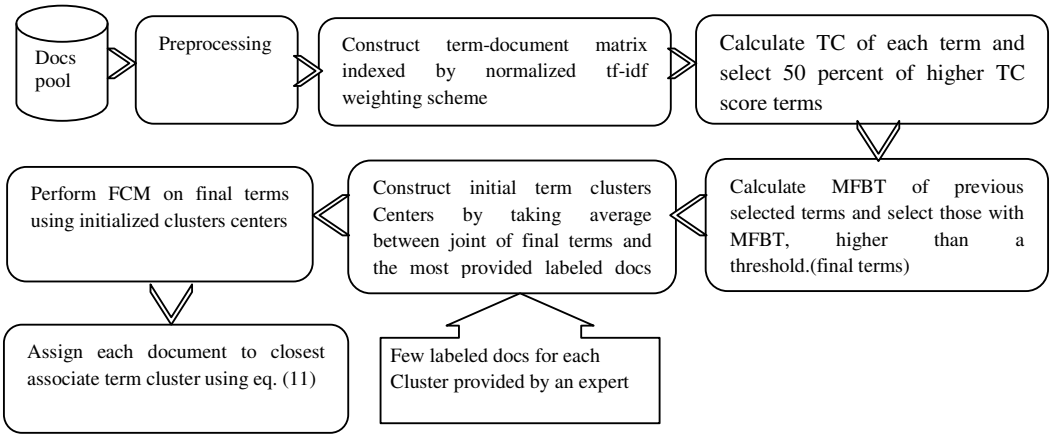

Figure 1. The proposed algorithm stages

5. EXPERIMENTAL SETUP

In this section, we use experimental results to show our proposed algorithm performance on several real-life datasets. The K-Means, spherical K-Means under the vector space model (VSM) and FCM clustering algorithms are used as comparison. For reminder, spherical K-Means uses the standard cosine measure, while standard K-Means uses the Euclidian distance for dissimilarity calculation for two samples.

5.1. Evaluation Measure

In this paper, we assume the number of clusters (i.e. *K)* is known, hence the number of clusters equals to the number of categories and can have a one-to-one correspondence between clusters and categories. We use Overall F-Measure and normalized Mutual Information (NMI) which are two common evaluation measures and frequently adopted in most of text mining researches. Both of them estimate the quality of clustering, given class label of the data.

5.1.1 Overall F-Measure

Since document category labels are known, we choose overall F-Measure which commonly is used in clustering evaluations, to asses our proposed document clustering algorithm. We note that labels aren't used in clustering process, except when a few labeled documents provided for centers clusters calculation. We have following definitions:

$$Precision_i = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (12)$$

$$Recall_i = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (13)$$

$$F - Measure_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (14)$$

$$Overall\ F - Measure = \frac{\sum_{i=1}^{k} N_i \times F - Measure_i}{\sum_{i=1}^{k} N_i} \quad (15)$$

Where N_i is actual size of class i. F-Measure combines precision and recall into a single number and overall F-Measure calculate overall F-Measure on all clusters with equal weights and don't bias toward the minority or majority classes. The clustering quality for overall F-Measure is from 0(worst) to 1(best).

5.1.2 Normalized Mutual Information (NMI)

NMI calculates amount of statistical information shared between two random variables. NMI demonstrates how much the clustering results are close to underling label distribution in data. Strehl et al [27] definition of NMI is as follow. Assume, *P* is a random variable that represents the clustering algorithm assignment on data and *Q* is a random variable represents the underlying class label of the data. NMI is defined as:

$$NMI = \frac{I(P;Q)}{(\frac{H(P)+H(Q)}{2})} \tag{16}$$

Where I*(P;Q)=H(P)-H(P|Q)* is the mutual information between two random variable *P* and *Q*, *H(P)* is entropy of *P* and *H(P|Q)* is conditional entropy of *P* given *Q*.

We use the NMI as follows:

Here, two random variables *P* and *Q* are $P = \{p_1, p_2, \dots, p_k\}$ and $Q = \{q_1, q_2, \dots, q_k\}$.

$$NMI = \frac{\sum_{p,q} n_{p,q} \log(\frac{n.n_{p,q}}{n_p n_q})}{\sqrt{\sum_p n_p \log(\frac{n_p}{n}) \sum_q n_q \log(\frac{n_q}{n})}} \tag{17}$$

Where n_p and n_q are the number of documents in cluster *p* and class q respectively. $n_{p,q}$ is the number of documents which either exist in cluster *p* and either in class *q*. NMI varies in [0 1] range and whatever be close to 1, means that clustering results are more match to original class labels of documents and whatever close to 0, means more randomness in clustering.

5.2. Datasets

As our first data set, we used 20-Newsgroup corpus collected by Lang [16]. This corpus is a collection of 20000 messages, collected from 20 different usenet newsgroups, each class contains 1000 messages. This collection has very noisy data. Spam, offensive and short messages for replies are frequent noisy data appeared in several categories. Moreover we noticed about 4.5 percent of documents had more than one label and removed all of them. Additionally, since some categories had very similar topics, we applied another test on newsgroup corpus and united the 5 "comp" categories, 3 "religion" categories, 3 "politics" categories, two "sport " categories and two "transportation" categories into 5 meta-categories. We named these two datasets as NG20 and NG10 respectively. We also used supervised Naïve Base (NB) classifier for evaluating how much our method results were close to supervised classification. For this, NB classifier trained over 1000 randomly chosen documents and tested on remaining. We perform this process in 10 fold and averaged the results.

The Reuters-21587 text categorization test dataset is a standard text categorization benchmark and contain 135 categories. We chose re0 and re1 dataset which are two skew subset of Reuters-21587.

These three dataset are good representation for different balance ratio. The balance ratio of a dataset is defined as ratio of the number of documents within smallest class to the number of documents in the largest class.

Table 1. Summary of datasets

Data Set	Source	#Doc	#Classes	#Words	Balance Ratio
Re0	Reuters21587	1504	13	11465	.018
Re1	Reuters21587	1657	25	3758	.026
NG20	20-Newsgroup	19949	20	43586	.991

5.3. Results

In the supervision help stage, we assumed that expert had enough skill to select best and related document for each category. We simulated this by random selecting 1% of documents from each category which contained at least 3 terms of final terms for NG20 and NG10 datasets. For re0 and re1, some minor classes which had less than 100 documents, we selected at most 5 documents for each category according to their size.

5.3.1 F-Measure

We founded m=1.2 as the best value for fuzziness level in all experiments. For re0 and re1 we chose those terms which their MFBT value were more than 3, and due to sparsity in minority classes added provided labeled document to unlabeled collection and didn't use their labels in clustering process. For comparison our results, we used Seeded K-Means and spherical Seeded K-Means which directly clusters documents on predefined k clusters. We seeded initial cluster centers for both algorithms by taking average between given labeled data and considered this fact that initial seeds should be noise-free. In the case of NG20 and NG10 which had enough samples in all categories, we selected terms as final terms which their MFBT value was more than 10. Moreover, for these two datasets we didn't use the seed samples in our evaluations.

The result of Overall F-Measure over NG20 and NG10 collections with several percent of available seed samples is shown in tables 2 and 3 respectively.

Table 2. Overall F-Measure results on NG20 dataset

Percentage of labeled data / Clustering methods	25%	50%	75%	100%
Seeded K-Means	.51±.08	.52±.05	.54±.05	.56±.02
Spherical Seeded K-Means	.53±.04	.55±.04	.56±.03	.58±.02
Our method	.60±.03	.62±.03	.65±.02	.67±.02
NB	.70±.01			

Table 3. Overall F-Measure results on NG10 dataset

Percentage of label data / Clustering methods	25%	50%	75%	100%
Seeded K-Means	.69±.04	.70±.03	.73±.03	.74±.02
Spherical Seeded K-Means	.73±.04	.75±.03	.77±.02	.78±.02
Our method	.78±.03	.8±.03	.82±.02	.83±.01
NB	.85±.01			

Due to sparse classes in re0 and re1, we used 5 labeled documents for classes which had less than 100 samples and up to 1% of classes size labeled documents for bigger classes. Overall F-Measure results on re0 and re1 in tables 4 and 5 shows our method performs well on skew datasets.

Table 4. Overall F-Measure results on re0 and re1text datasets

Datasets / Clustering Methods	re0	re1
Seeded K-Means	.61±.03	.74±.02
Spherical Seeded K-Means	.63±.02	.77±.03
Our method	.67±.03	.81±.03

5.3.2 NMI

Addition to F-Measure, we utilized the NMI in order to evaluate the role of seed fraction for initialization center of term clusters in all algorithm over all datasets. Also, we ran the K-Means

with the random seeding over all datasets to assess how much supervision help can be effective on results.

We created a learning curve with 10-fold cross-validation for each dataset. At first, divided each dataset to 10 equal fold and for studying the effect of seeding, one fold set aside for test and remaining data set for training. Then we selected seeds from training set by varying the seed fraction from 0 to 1 by steps of .1 and result at each point of curve was obtained by averaging over 10 folds. The several clustering algorithms performed on the whole dataset, but NMI measure was calculated on test sets in each folds.

As shown in figures 2 and 3, we can see that the semi-supervised (Seeded K-Means, Sp K-Means and our method) learning, without fluctuation perform better than the unsupervised (Random K-Means) learning in terms of NMI measure. Spherical K-Means performs a bit better than Seeded K-Means, for the sake of use cosine similarity instead of Euclidian distance. An notable point in two following figures is that all three semi-supervised algorithm have a long leap in the first seed fraction and low increment leaps in next fractions. This point is more obvious in our method. Our method reason is hidden in final terms which generated by MFBT in the second feature selection phase. The seeds were those documents which contained some of these salient final terms. This causes to all seeds be in a same importance level and the first fraction of them has the most portion in results.

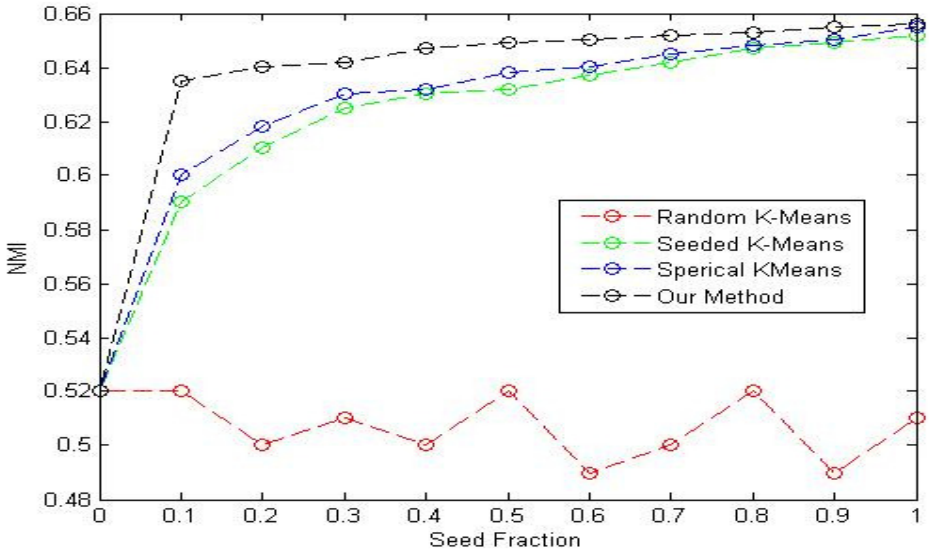

Figure 2. Comparison of NMI values on full 20newsgroup (NG20) with increasing in seed fraction

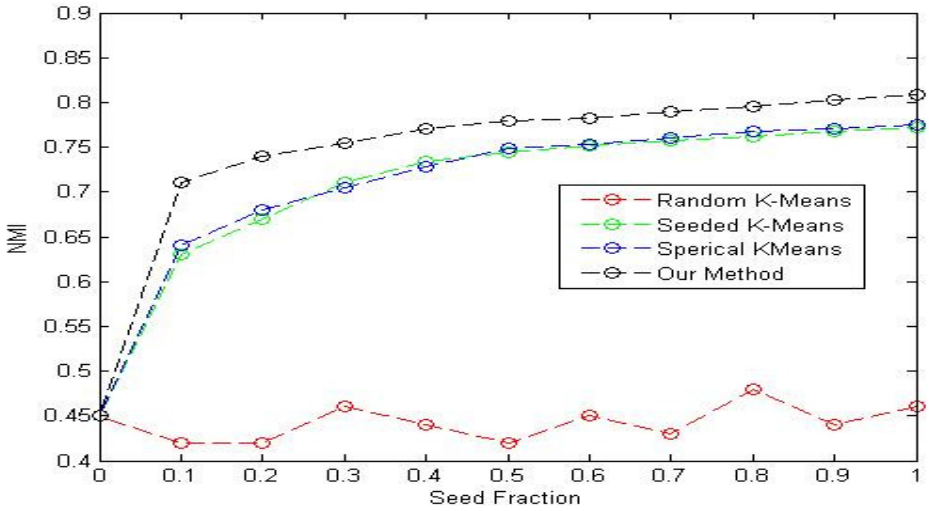

Figure 3. Comparison of NMI values on reduced newsgroup (NG10) with increasing in seed fraction

The NMI learning curve for re0 and re1 datasets is shown in figures 4 and 5 respectively. The problem in first fractions for skew datasets like re0 and re1 -which has a few documents in some categories-, is that maybe seeds not specified for some clusters. Figures 4 and 5 show that the NMI measure doesn't decrease substantially for the random seeding for small size clusters.

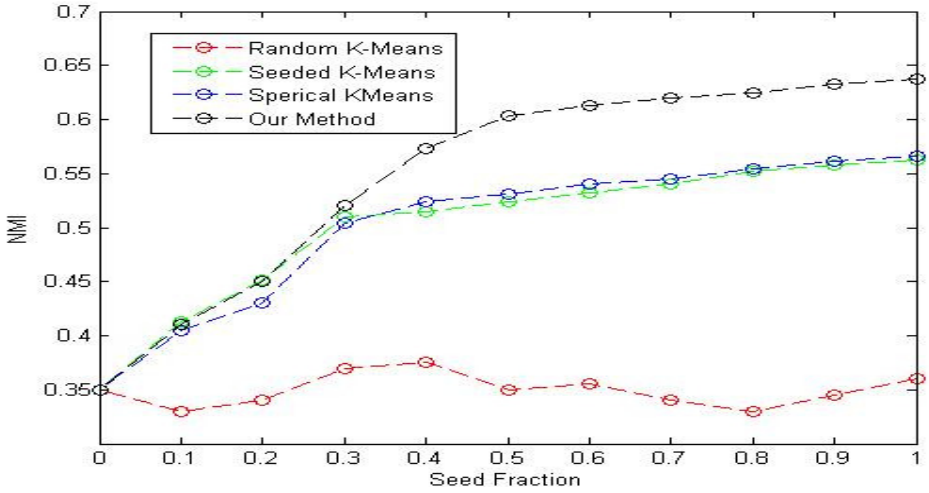

Figure 4. Comparison of NMI values on incomplete seeding re0 with increasing in seed fraction

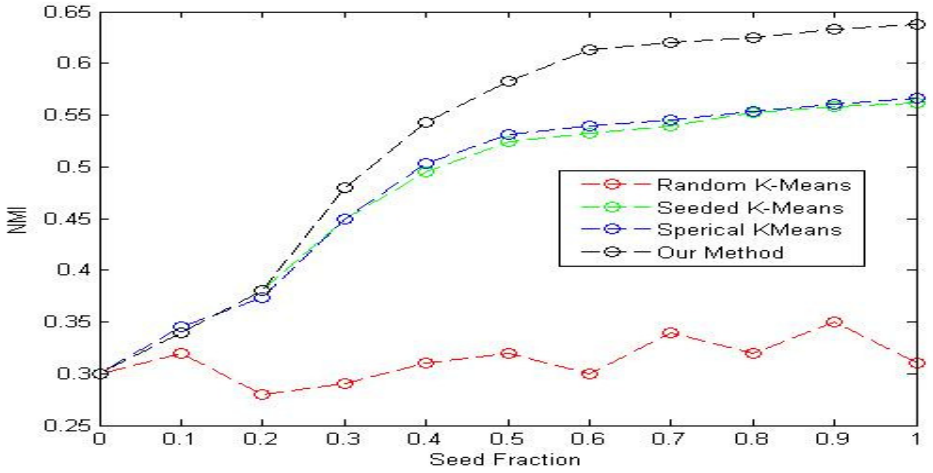

Figure 5. Comparison of NMI values on incomplete seeding re1 with increasing in seed fraction

In the two first fractions, three semi-supervised algorithms haven't long leaps like NG20 and NG10 learning curves (Figures 2 and 3), for the sake of random seeding in some small size clusters. In the third and next fractions we can see, the curves close to their real values.

6. CONCLUSION

Our main goal in the proposed method is to use term clustering over certain and free of noise sample space for document clustering. In our idea, directly clustering of documents, causes to deviate the some parameters such as cluster centers from their real values in the clustering process, for the sake of existence of many noisy documents. We selected a small and perfect set of terms (features) and initialized term cluster centers in a semi-supervised fashion and used the c-means algorithm as the main clustering algorithm to fuzzify the membership value of terms to all clusters. Our motivation for utilizing C-Means was that any term can't crisply belong to one subject (cluster), especially in datasets with the close semantic relation categories. Experimental results on several text datasets showed that the term clustering can significantly improve the document clustering results on condition the selected terms be perfect and free of noise. We hope to incorporate the unsupervised clustering methods in our future works in order to initialize cluster centers instead of supervision help.

REFERENCES

[1] M. Aono and H. Doi, "A method for query expansion using a hierarchy of clusters," In AIRS, pages 479–484, 2005.

[2] J. Uszkoreit and T. Brants, "Distributed word clustering for large scale class-based language modeling in machine translation," In ACL International Conference Proceedings, Columbus, OH, USA, 2008.

[3] W. Chen, X. Chang, H. Wang, J. Zhu, and T. Yao," Automatic word clustering for text categorization using global information," In AIRS Internation Conference Proceedings, volume 3411 of Lecture Notes in Computer Science, pages 1–11. Springer, 2004.

[4] C. Samuelsson and W. Reichl, " A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics," In IEEE ICASSP International Conference Proceedings. IEEE Computer Society, 1999.

[5] S. St ¨ uker, J. Fay, and K. Berkling, "Towards context-dependent phonetic spelling error correction in childrens freely composed text for diagnostic and pedagogical purposes," in Interspeech 2011, 2011.

[6] V. Hodge and J. Austin, "Hierarchical word clustering automatic thesaurus generation,". Neuro computing, 48:819–846,2002.

[7] P. Brown, V. Pietra, P. Souza, J. Lai, and R. Mercer, "Class-based n-gram models of natural language," Computational linguistics, 18(4):467–479, 1992.

[8] S. Momtazi and D. Klakow. 2009, "A word clustering approach for language model-based sentence retrieval in question answering systems," In Proc. of ACMCIKM.

[9] Dagan, L. Lee, and F. C. N. Pereira. 1999, "Similarity-based models of word cooccurrence probabilities," Machine Learning, 34(1-3):43–69.

[10] Pereira, F. C. N., Tishby, N., &Lee, L. (1993). Distributional clustering of English words. In 31st AnnualMeeting of the ACL(p. 183-190). Somerset, New Jersey: Association for Computational Linguistics. (Distributed by Morgan Kaufmann, San Francisco)

[11] N. Slonim and N. Tishby. The power of word clusters for text classification. In 23rd European Colloquium on Information Retrieval Research, 2001.

[12] N. Tishby, F.C. Pereira and W. Bialek. The Information Bottleneck Method In Proc. of the 37-th Allerton Conference on Communication and Computation, 1999.

[13] M.-R. Amini and N. Usunier, A contextual query expansion approach by term clustering for robust text summarization. In Proceedings of DUC, 2007

[14] Salton, G. (1989). "Automatic Text Processing" The Transformation, Analysis, and Retrieval of Information by Computer. Addison-wesley, Reading, Pennsylvania.

[15] Dhillon, I. S., & Modha, D. S. (2001), "Concept decompositions for large sparse text data using clustering. Machine Learning," 42, 143–175.

[16] K. Lang. "Learning to filter netnews". In Proc. of the 12th Int. Conf. on Machine Learning, 1995.

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

[18] Yang, Y., & Pedersen, J. O. (1997), "A comparative study on feature selection in text categorization," Proc. of ICML-97 (pp. 412-420).

[19] Yang, Y. (1995), "Noise reduction in a statistical approach to text categorization," Proc. of SIGIR'95 (pp. 256-263).

[20] Dash, M., & Liu, H. (2000). Feature Selection for Clustering. Proc. of PAKDD-00 (pp. 110-121).

[21] LIU, T., LIU, S., CHEN, Z., AND MA, W.-Y. 2003, "An evaluation on feature selection for text clustering." In Proceedings of the 20th International Conference on Machine Learning, August 21–24, T. Fawcett and N. Mishra, Eds. AAAI Press, 488–495.

[22] Jashki, M.A.; Makki, M.; Bagheri, E. & Ghorbani, A. (2009). An iterative hybrid filter-wrapper approach to feature selection for document clustering. Advances in Artificial Intelligence, LNS Vol. 5549, pp. 74-85.

[23] Galavotti, L., Sebastiani, F., & Simi, M. (2000). Feature selection and negative evidence in automated text categorization. Proc. of KDD-00.

[24] Unsupervised Feature Subset Selection Nicolaj Søndberg-Madsen.

[25] Valente de Oliveira, J., Pedrycz, W., "Advances in Fuzzy Clustering and its Applications," John Wiley & Sons, pp 3-30, 2007.

[26] Guyon and A. Elissee , An introduction to variable and feature selection, The Journal of Machine Learning Research 3 (2003), 1157-1182.

[27] Strehl, A., & Ghosh, J. (2002). Cluster ensembles — a knowledge reuse framework for combining partitions. Journal of Machine Learning Research, 3, 583–617.