An Evaluation on Feature Selection for Text Clustering

Tao Liu LTMAILBOX@263.SINA.COM
Department of Information Science, Nankai University, Tianjin 300071, P. R. China

Shengping Liu LSP@IS.PKU.EDU.CN
Department of Information Science, Peking University, Beijing 100871, P. R. China

Zheng Chen ZHENGC@MICROSOFT.COM
Wei-Ying Ma WYMA@MICROSOFT.COM
Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P. R. China

Abstract

Feature selection methods have been successfully applied to text categorization but seldom applied to text clustering due to the unavailability of class label information. In this paper, we first give empirical evidence that feature selection methods can improve the efficiency and performance of text clustering algorithm. Then we propose a new feature selection method called "Term Contribution (TC)" and perform a comparative study on a variety of feature selection methods for text clustering, including Document Frequency (DF), Term Strength (TS), Entropy-based (En), Information Gain (IG) and $\aleph^2$ statistic (CHI). Finally, we propose an "Iterative Feature Selection (IF)" method that addresses the unavailability of label problem by utilizing effective supervised feature selection method to iteratively select features and perform clustering. Detailed experimental results on Web Directory data are provided in the paper.

1. Introduction

Text clustering is one of the central problems in text mining and information retrieval area. The task of text clustering is to group similar documents together. It had been applied to several applications, including improving retrieval efficiency of information retrieval systems (Kowalski, 1997), organizing the results returned by a search engine in response to user's query (Zamir et al., 1997), browsing large document collections (Cutting et al., 1992), and generating taxonomy of web documents (Koller & Sahami, 1997), etc.

In text clustering, a text or document is always represented as a bag of words. This representation raises one severe problem: the high dimensionality of the feature space and the inherent data sparsity. Obviously, a single document has a sparse vector over the set of all terms. The performance of clustering algorithms will decline dramatically due to the problems of high dimensionality and data sparseness (Aggrawal &Yu, 2000). Therefore it is highly desirable to reduce the feature space dimensionality. There are two commonly used techniques to deal with this problem: feature extraction and feature selection. Feature extraction is a process that extracts a set of new features from the original features through some functional mapping (Wyse et al., 1980), such as principal component analysis (PCA) (Jolliffe, 1986) and word clustering (Slonim & Tishby, 2000). The feature extraction methods have a drawback that the generated new features may not have a clear physical meaning so that the clustering results are difficult to interpret (Dash & Liu, 2000).

Feature selection is a process that chooses a subset from the original feature set according to some criterions. The selected feature retains original physical meaning and provides a better understanding for the data and learning process. Depending on if the class label information is required, feature selection can be either unsupervised or supervised. For supervised methods, the correlation of each feature with the class label is computed by distance, information dependence, or consistency measures (Dash & Liu, 1997). Further theoretical study based on information theory can be found on (Koller & Sahami, 1996) and complete reviews can be found on (Blum & Langley, 1997; Jain et al., 2000; Yang & Pedersen, 1997).

As for feature selection for clustering, there have some works on it. Firstly, any traditional feature selection method that does not need the class information, such as document frequency (DF) and term strength (TS) (Yang, 1995), can be easily applied to clustering. Secondly, there are some newly proposed methods, for example, entropy-based feature ranking method (En) is proposed by Dash and Liu (2000) in which feature importance is measured by the

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

contribution to an entropy index based on the data similarity; the individual "feature saliency" is estimated and an Expectation-Maximization (EM) algorithm using Minimum Message Length criteria is derived to select the feature subset and the number of clusters (Martin et al., 2002).

While the methods mentioned above are not directly targeted to clustering text documents, in this paper we introduce two novel feature selection methods for text clustering. One is Term Contribution (TC) which ranks the feature by its overall contribution to the documents similarity in a dataset. Another is Iterative Feature Selection (IF), which utilizes some successful feature selection methods (Yang & Pedersen, 1997), such as Information Gain (IG) and χ^2 statistics (CHI), to iteratively select features and perform text clustering at the same time.

Another contribution of this paper is a comparative study on feature selection for text clustering. We investigate (a) to what extent feature selection can improve the clustering quality, (b) how much of the document vocabulary can be reduced without losing useful information in text clustering, (c) what the strengths and weaknesses of existing feature selection methods are when applied to text clustering, and (d) what the difference is among the results of different datasets. In this paper, we try to address these problems by empirical evidences. We first show that feature selection methods can improve the efficiency and performance of text clustering in ideal cases, in which the class label for each document is already known. Then we perform a comparative study on various feature selection methods for text clustering. Finally, we evaluate the performance of iterative feature selection method based on K-means using entropy and precision measures.

The rest of this paper is organized as follows. In Section 2, we give a brief introduction on several feature selection methods and propose a new feature selection method, called Term Contribution. In Section 3, we propose a new iterative feature selection method that utilizes the supervised feature selection algorithm without the need to know the class information in advance. In Section 4, we conduct several experiments to compare the effectiveness of different feature selection methods in ideal and real cases. Finally, we summarize our major contributions in Section 5.

2. Feature Selection Methods

In this Section, we give a brief introduction on several effective feature selection methods, including two supervised methods, IG and CHI, and four unsupervised methods, DF, TS, En and TC. All these methods assign a score to each individual feature and then select features which are greater than a pre-defined threshold.

In the following, let D denote the documents set, M the dimension of the features, and N the number of documents in the dataset.

2.1 Information Gain (IG)

Information gain (Yang & Pedersen, 1997) of a term measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document. Let m be the number of classes. The information gain of a term t is defined as

$$\begin{aligned} IG(t) = &-\sum_{i=1}^{m} p(c_i)\log p(c_i) \\ &+ p(t)\sum_{i=1}^{m} p(c_i \mid t)\log p(c_i \mid t) \\ &+ p(\bar{t})\sum_{i=1}^{m} p(c_i \mid \bar{t})\log p(c_i \mid \bar{t}) \end{aligned} \quad (1)$$

2.2 χ^2 statistic (CHI)

The χ^2 statistic measures the association between the term and the category (Galavotti et al., 2000). It is defined to be

$$\chi^2(t,c) = \frac{N \times (p(t,c) \times p(\bar{t},\bar{c}) - p(t,\bar{c}) \times p(\bar{t},c))^2}{p(t) \times p(\bar{t}) \times p(c) \times p(\bar{c})} \quad (2)$$

$$\chi^2(t) = \underset{i=1}{\overset{m}{avg}}\{\chi^2(t,c_i)\} \quad (3)$$

2.3 Document Frequency (DF)

Document frequency is the number of documents in which a term occurs in a dataset. It is the simplest criterion for term selection and easily scales to a large dataset with linear computation complexity. It is a simple but effective feature selection method for text categorization (Yang & Pedersen, 1997).

2.4 Term Strength (TS)

Term strength is originally proposed and evaluated for vocabulary reduction in text retrieval (Wilbur & Sirotkin, 1992), and later applied to text categorization (Yang, 1995). It is computed based on the conditional probability that a term occurs in the second half of a pair of related documents given that it occurs in the first half:

$$TS(t) = p(t \in d_j \mid t \in d_i), d_i, d_j \in D \bigcap sim(d_i, d_j) > \beta \quad (4)$$

where β is the parameter to determine the related pairs. Since we need to calculate the similarity for each document pair, the time complexity of TS is quadratic to the number of documents. Because the class label information is not required, this method is also suitable for term reduction in text clustering.

2.5 Entropy-based Ranking (En)

Entropy-based ranking is proposed by Dash and Liu (2000). In this method, the term is measured by the entropy reduction when it is removed. The entropy is defined as the equation (5):

$$E(t) = -\sum_{i=1}^{N}\sum_{j=1}^{N}(S_{i,j}\times\log(S_{i,j}) + (1-S_{i,j})\times\log(1-S_{i,j})), \quad (5)$$

where $S_{i,j}$ is the similarity value between the document d_i and d_j. $S_{i,j}$ is defined as the equation (6):

$$S_{i,j} = e^{-\alpha\times dist_{i,j}}, \alpha = -\frac{\ln(0.5)}{\overline{dist}} \quad (6)$$

where $dist_{i,j}$ is the distance between the document d_i and d_j after the term *t* is removed, $\overline{dist}$ is the average distance among the documents after the term *t* is removed.

The most serious problem of this method is its high computation complexity $O(MN^2)$. It is impractical when there is a large number of documents and terms, and therefore, sampling technique is used in real experiments (Dash & Liu, 2000).

2.6 Term Contribution (TC)

We introduce a new feature selection method called "Term Contribution" that takes the term weight into account. Because the simple method like DF assumes that each term is of same importance in different documents, it is easily biased by those common terms which have high document frequency but uniform distribution over different classes. TC is proposed to deal with this problem,

The result of text clustering is highly dependent on the documents similarity. So the contribution of a term can be viewed as its contribution to the documents' similarity. The similarity between documents d_i and d_j is computed by dot product:

$$sim(d_i, d_j) = \sum_t f(t, d_i)\times f(t, d_j) \quad (7)$$

where $f(t,d)$ represents the tf*idf (Salton, 1989) weight of term t in document d.

So we define the contribution of a term in a dataset as its overall contribution to the documents' similarities. The equation is

$$TC(t) = \sum_{i,j\mid i\neq j} f(t, d_i)\times f(t, d_j) \quad (8)$$

"ltc" scheme is used to compute each term's tf*idf value which takes the log of term's frequency in the document, then multiplies it by the IDF weight of this term, and finally normalizes the document length.

If the weights of all terms are equal, we simply set $f(t,d)=1$ when the term t appears in the document d. Then the value *TC(t)* can be written as the equation (9):

$$TC(t) = DF(t)(DF(t)-1) \quad (9)$$

Since the transformation is increasing monotonously while the *DF(t)* (the document frequency of the term *t*) is a positive integer, DF is just the special case of TC.

Using the inverse document indexing technology (Salton, 1989), the time complexity of TC is $O(M\overline{N}^2)$, where $\overline{N}$ is the average documents per term occurs in.

3. Iterative Feature Selection (IF) Method

Feature selection methods have successfully applied to text categorization for long years. Feature selection can dramatically improve the efficiency of text categorization algorithm by removing up to 98% unique terms and even improve the categorization accuracy to some extent (Yang & Pedersen, 1997). So it is an interesting idea to apply feature selection methods to text clustering task to improve the clustering performance. In order to test our idea, several experiments were conducted. As can be seen from Section 4.3.1 and Section 4.3.2, if the class label information is known, supervised feature selection methods, such as CHI and IG, are much more efficient than unsupervised feature selection methods for text clustering, not only more terms can be removed, but also better performance can be yielded.

Supervised feature selection method can not be directly applied to text clustering because of the unavailability of the required class label information. Fortunately, we found that clustering performance and feature selection can be reinforced by each other. On the one hand, good clustering results will provide good class labels to select better features for each category; on the other hand, better features will help clustering to improve the performance to provide better class labels. Enlightened by the EM algorithm, we propose a novel iterative feature selection method, which utilizes supervised feature selection methods for text clustering methods.

EM is a class of iterative algorithms for maximum likelihood estimation in problems with incomplete data (Dempster et al., 1977). Suppose we have a dataset generated from a model with parameters and the data has missing values or unobservable (hidden) variables. To find the parameters that can maximize the likelihood function, EM algorithm first calculates the expectation of hidden variables based on the current parameters estimation at the E-step. Then at the M-step, missing values are replaced by the expected values calculated in the E-step to find a new estimation of parameters that can maximize the complete data likelihood function. These two steps are iterated to convergence.

In the clustering applications, it is normally assumed that a document is generated by a finite mixture model and there is one-to-one correspondence between mixture components and clusters. So the likelihood function $p(D\mid\theta)$, i.e. the probability of all documents D given the model parameter θ, can be written as the equation (10):

$$p(D\mid\theta) = \prod_{i=1}^{N}\sum_{j=1}^{|C|} p(c_j\mid\theta)p(d_i\mid c_j,\theta) \quad (10)$$

where c_j is j-th cluster, $|C|$ is the number of clusters, $p(c_j \mid \theta)$ is the prior distribution of cluster *j*, and $p(d_i \mid c_j, \theta)$ is the distribution of document *i* in cluster *j*. It is further assumed that the terms are conditional independent given the class label, and then the likelihood function can be rewritten as the equation (11):

$$p(D \mid \theta) = \prod_{i=1}^{N} \sum_{j=1}^{|C|} (p(c_j \mid \theta) \prod_{t \in d_i} p(t \mid c_j, \theta)) \tag{11}$$

where $p(t \mid c_j, \theta)$ is term distribution for the term *t* in the cluster *j*. Not all terms are equivalently relevant to the document, so $p(t \mid c_j, \theta)$ can be treated as the weighted sum of relevant distribution and irrelevant distribution as the equation (12):

$$\begin{aligned} p(t \mid c_j, \theta) = z(t)\, p(t \text{ is relevant} \mid c_j, \theta) \\ + (1 - z(t))\, p(t \text{ is irrelevant} \mid \theta) \end{aligned} \tag{12}$$

where $z(t) = p(t \text{ is relevant})$ is defined as the probability that the term *t* is relevant. In addition, if the term is not relevant, the term distribution is assumed to be the same over different clusters and is denoted as $p(t \text{ is irrelevant} \mid \theta)$. Hence, the likelihood function can be written as the equation (13):

$$\begin{aligned} p(D \mid \theta) = \prod_{i=1}^{N} \sum_{j=1}^{|C|} (p(c_j \mid \theta) \\ \prod_{i \in d_i} (z(t)\, p(t \text{ is relevant} \mid c_j, \theta) + (1 - z(t))\, p(t \text{ is irrelevant} \mid \theta))) \end{aligned} \tag{13}$$

To maximize this likelihood function, EM algorithm can find a local maximum by iterating the following two steps:

(1) E-step: $\hat{z}^{(k+1)} = E(z \mid D, \hat{\theta}^{(k)})$ (14)

(2) M-step: $\hat{\theta}^{(k+1)} = \arg\max_{\theta}\ p(D \mid \theta, \hat{z}^{(k)})$ (15)

The E-step corresponds to calculating the expected feature relevance given the clustering result, and the M-step corresponds to calculating a new maximum likelihood estimation of the clustering result in the new feature space.

The proposed EM algorithm for text clustering and feature selection is a general framework and the full implementation is beyond the scope of this paper. Our Iterative Feature Selection method can be fitted into this framework. On the E-step, to approximate the expectation of feature relevance, we use supervised feature selection algorithm to calculate the relevance score for each term, then the probability for the term relevance is simplified to $z(t) = \{0,1\}$ according to whether the term relevance score is larger than a predefined threshold value. So at each iteration, we will remove some irrelevant terms based on the calculated relevance of each term. On the M-step, because K-means algorithm can be described by slightly extending the mathematics of the EM algorithm to the hard threshold case (Bottou et al., 1995), we use K-means clustering algorithm to obtain the cluster results based on the selected terms.

4. Experiments

We first conducted an ideal case experiment to demonstrate that supervised feature selection methods, including IG and CHI, can significantly improve the clustering performance. Then, we evaluated the performance of unsupervised feature selection methods, including DF, TS, TC and En in real case. Finally we evaluated the iterative feature selection algorithm. K-means was chosen as our basic clustering algorithm and entropy measure and precision measure were used to evaluate the clustering performance. Since K-means clustering algorithm is easily influenced by selection of initial centroids, we random produced 10 sets of initial centroids for each data set and averaged 10 times performances as the final clustering performance. Before performing clustering, tf*idf (with "ltc" scheme) was used to calculate the weight of each term.

4.1 Performance Measures

Two kinds of popular measurements, entropy and precision, were used to evaluate the clustering performance.

4.1.1 Entropy

Entropy measures the uniformity or purity of a cluster. Let G', G denote the number of obtained clusters and the number of original classes respectively. Let *A* denote the set of documents in a obtained cluster, and the class label of each document $d_i \in A, i = 1,...,|A|$ is denoted as $label(d_i)$, which takes value $c_j (j = 1...,G)$. The entropy for all clusters is defined by the weighted sum of the entropy for all clusters, as shown in the equation (16):

$$Entropy = -\sum_{k=1}^{G'} \frac{|A_k|}{N} \sum_{j=1}^{G} p_{jk} \times \log(p_{jk}) \tag{16}$$

where

$$P_{jk} = \frac{1}{|A_k|} |\{d_i \mid label(d_i = c_j)\}| \cdot \tag{17}$$

4.1.2 Precision

Since entropy is not intuitive, we choose another measure to evaluate the clustering performance, i.e. precision. For each cluster, it always consists of documents from several different classes. So we simply choose the class label which shares with most documents in this cluster as the final class label. Then, the precision for each cluster is defined as:

$$Precision(A) = \frac{1}{|A|}\max(|\{d_i \mid label(d_i) = c_j\}|) \quad (18)$$

In order to avoid the possible bias from small clusters which have very high precision, the final precision is defined by the weighted sum of the precision for all clusters, as shown in the equation (19):

$$Precision = \sum_{k=1}^{G'} \frac{|A_k|}{N} \Pr ecision(A_k) \quad (19)$$

4.2 Data Sets

As reported in past research works (Yang & Pederen, 1997), text categorization performance varies greatly on different dataset. So we chose three different text datasets to evaluate text clustering performance, including two standard labeled datasets: Reuters-21578 [1] (Reuters), 20 Newsgroups [2] (20NG), and one web directory dataset (Web) collected from the Open Directory Project[3]. There were total 21578 documents in Reuters, but we only chose the documents that have at least one topic and Lewis split assignment, and assigned the first topic to each document before the evaluation. The information about these datasets is shown in Table 1.

Table 1. The three datasets properties

DATA SETS	CLASSES NUM.	DOCS NUM.	TERMS NUM	AVG. TERMS PER DOC	AVG. DF PER TERM
REUTERS	80	10733	18484	40.7	23.6
20NG	20	18828	91652	85.3	17.5
WEB	35	5035	56399	131.9	11.8

4.3 Results and Analysis

4.3.1 SUPERVISED FEATURE SELECTION

First, we conducted an ideal case experiment to see whether good terms can help text clustering. That is, we applied supervised feature selection methods to choose the best terms based on the class label information. Then, we executed the text clustering task on these selected terms and compared the clustering results with the baseline system, which clustered the documents on full feature space.

The entropy comparison over different datasets is shown in Figure 1, and the precision comparison is shown in Figure 2. As can be seen, at least 90% terms can be removed with either an improvement or no loss in clustering performance on any dataset. With more terms removed, the clustering performances on Reuters and 20NG changes a little, but on Web Directory dataset, there is a significant performance improvement. For example, using CHI method on Web Directory dataset, when 98% terms are removed, the entropy is reduced from 2.305 to 1.870 (18.9% entropy reduction relatively), the precision is improved from 52.9% to 63.0% (19.1% precision improvement relatively). Certainly, the results are just the upper bound of the clustering performance in real case because it is difficult to select discriminative terms without prior class labels information.

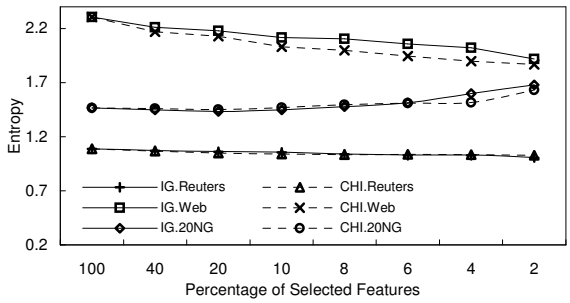

Figure 1. Entropy comparison on 3 datasets (supervised)

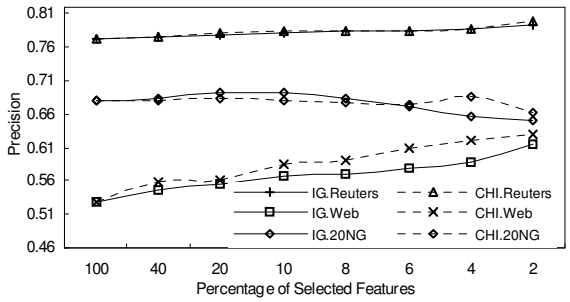

Figure 2. Precision comparison on 3 datasets (supervised)

Feature selection makes little progress on Reuters and 20NG, while achieves much improvement on Web directory dataset. It motivates us to find out the reason. As reported in the past research works on feature selection for text categorization, the classification accuracy is almost the same after removing some terms from Reuters and 20NG datasets with most classifiers, including Naïve Bayesian classifier and KNN classifier (Yang & Pedersen, 1997). We conclude that most terms in these two datasets still have discriminative values for classification although few terms per class are enough to achieve acceptable classification accuracy. In other words, there are few noisy terms in these two datasets. Similarly, although feature selection can reduce the dimension of the feature, the clustering performance can not be improved significantly because some useful terms are also ignored after removing some terms from these two datasets. However, it is different for the Web Directory data, in which there are much more

[1] http://www.daviddlewis.com/resources/testcollections/
[2] http://www.ai.mit.edu/people/jrennie/20Newsgroups/
[3] http://dmoz.org/

noisy terms. To prove this, we conducted a Naïve Bayesian classification experiment on Web Directory dataset and found that the classification accuracy increased from 49.6% to 57.6% after removing 98% terms. Hence, when these noisy terms, such as "site, tool, category", are removed in clustering, the clustering performance can also be significantly improved.

4.3.2 UNSUPERVISED FEATURE SELECTION

The second experiment we conducted is to compare the unsupervised feature selection methods (DF, TS, TC, and En) with supervised feature selection methods (IG and CHI).

The entropy and precision results on Reuters are shown in Figure 3 and Figure 4.

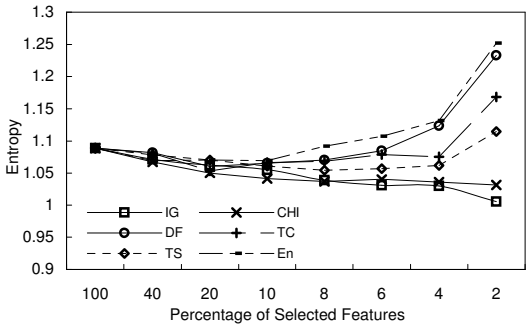

Figure 3. Entropy comparison on Reuters (unsupervised)

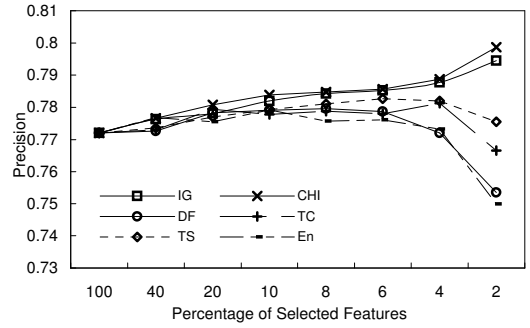

Figure 4. Precision comparison on Reuters (unsupervised)

From these figures, we found the following points. First, unsupervised feature selection can also improve the clustering performance when a certain terms are removed. For example, any unsupervised feature selection methods can achieve about 2% entropy reduction and 1% precision improvement relatively while 90% terms are removed. Second, unsupervised feature selection can be comparable to supervised feature selection with up to 90% term removal. When more terms are removed, the performances of supervised methods can be still improved, but the performances of unsupervised methods drop quickly. In order to find out the reason, we compared the terms selected by IG and TC at different removal thresholds. After analysis, we found that at the beginning stage, low document frequency terms were removed by both methods, but at the next stage, with more terms removed, the discrimination value between the term and the class is much more important than the document frequency for term selection. Since discrimination value is dependent on the class information, supervised method IG could keep those less common but discriminative terms, such as "OPEC, copper, drill". However, without class information, TC could not decide whether a term was discriminative and still kept those common terms, such as "April, six, expect". Hence, it is obvious that supervised methods are much better than unsupervised methods when more terms are removed.

Finally, compared with other unsupervised feature selection methods, TC is better than DF and En, but little worse than TS. For example, as Figure 3 shows, when 96% terms are removed, the entropies of DF and En are much higher than the baseline value (on full feature space) but the entropies of TC and TS are still lower than the baseline value. En is very similar to DF because the removal of common term is easily to cause more entropy reduction than the removal of rare term. TS is sensitive to the document similarity threshold β. When β is just suitable, the performance of TS is better than TC. However, the threshold is difficult to tune. In addition, the time complexity of TS (at least $O(N^2)$) is always higher than that of TC ($O(M\overline{N}^2)$), because $\overline{N}$ is always much small than N (see Table 1). Therefore, TC is the preferred method with effective performance and low computation cost.

Similar points can be found on the rest two datasets (20NG and Web) (Figure 5 and Figure 6). Due to the limitations of the paper length, we only drew the entropy measure for these two datasets. As can be seen from these figures, the best performances are also yielded on Web Directory data. About 8.5% entropy reduction and 8.1% precision improvement are achieved while 96% features are removed with TS method.

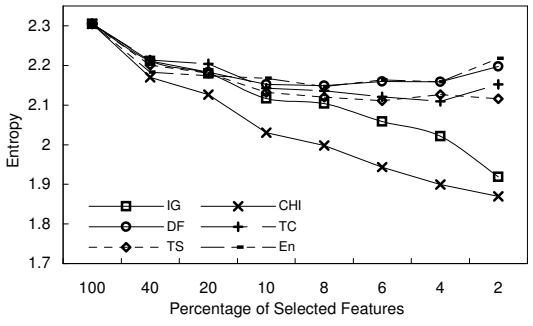

Figure 5. Entropy comparison on Web Directory (unsupervised)

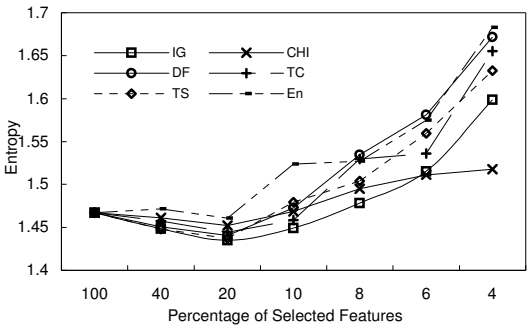

Figure 6. Entropy comparison on 20NG (unsupervised)

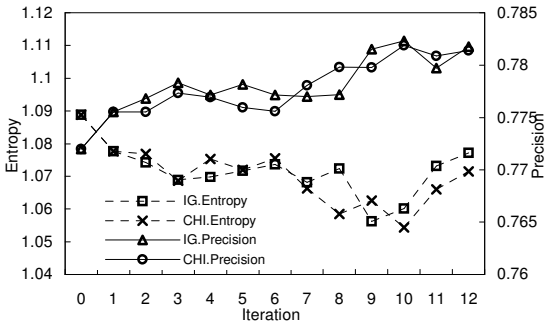

Figure 7. Entropy & precision with IF selection on Reuters

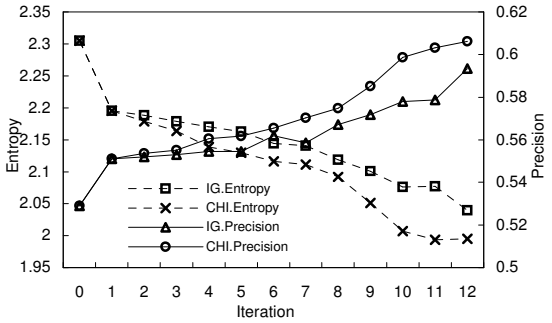

Figure 8. Entropy & precision with IF selection on Web Directory

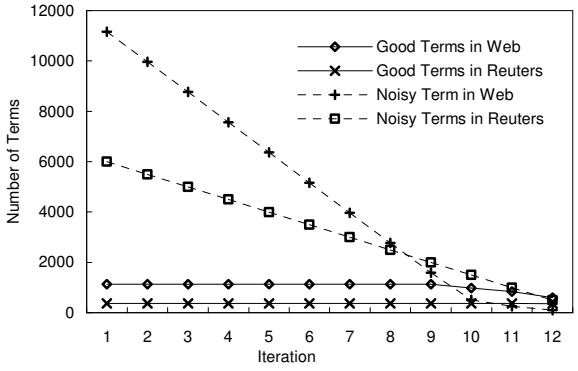

Figure 9. Feature Tracking on Reuters & Web Directory

4.3.3 ITERATIVE FEATURE SELECTION

The third experiment we conducted is to measure the iterative feature selection algorithm proposed in Section 3. Since IG and CHI have good performance in ideal case, they were chosen in iterative feature selection experiment. In order to speed up the iterative process, we removed the terms whose document frequency was lower than 3 before clustering. Then at each iteration, about 10% terms (or 3% terms when less than 10% terms left) with lowest ranking score outputted by IG or CHI were removed out. Figure 7 displays the entropy (dashed line) and precision (solid line) results on Reuters. Figure 8 displays the results on Web Directory dataset. We did not show the results on 20NG due to the limitations of page.

As can be seen from Figure 7 and Figure 8, the performance of iterative feature selection is quite good. It is very close to the ideal case and much better than any unsupervised feature selection methods. For example, after 11 iterations with CHI selection on Web Directory dataset (nearly 98% terms removal), the entropy was reduced from 2.305 to 1.994 (13.5% entropy reduction relatively) and the precision was improved from 52.9% to 60.6% (14.6% precision improvement relatively). It is close to the upper bound from the ideal case (18.9% entropy reduction and 19.1% precision improvement, see Section 4.3.1).

In order to verify that iterative feature selection works, we traced the terms which were filtered out at each iteration. Firstly, we assumed that the top 2% terms ranked by CHI in ideal case experiment were good terms, and others were noisy terms. Then we calculated how many "good" terms were kept at each iteration. As can be seen from Figure 9, the "good" terms (two bottom lines) are almost kept and most noisy terms (two top lines) are filtered out after 10 iterations.

5. Conclusions

In this paper, we first demonstrated that feature selection can improve the text clustering efficiency and performance in ideal case, in which features are selected based on class information. But in real case the class information is unknown, so only unsupervised feature selection can be exploited. In many cases, unsupervised feature selection are much worse than supervised feature selection, not only less terms they can remove, but also much worse clustering performance they yield. In order to utilize the efficient supervised methods, we proposed an iterative feature selection method that iteratively performs clustering and feature selection in a unified framework. It is found that its

performance is close to the ideal case and much better than any unsupervised methods. Another work we have done is a comparative study on several unsupervised feature selection methods, including DF, TS, En, and a new proposed method TC. It is found that TS and TC are better than DF and En. Since TS has high computation complexity and is difficult to tune its parameters, TC is the preferred unsupervised feature selection method for text clustering.

References

Aggrawal, C.C., & Yu, P.S. (2000). Finding generalized projected clusters in high dimensional spaces. *Proc. of SIGMOD'00* (pp. 70-81).

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2001). On Feature Distributional Clustering for Text Categorization. *Proc. of SIGIR'01* (pp. 146-153).

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *1(2)*, 245-271.

Bottou L., & Bengio Y. (1995). Convergence properties of the k-means algorithms. *Advances in Neural Information Processing Systems,* 7, 585-592.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey.W. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proc. of SIGIR'92* (pp. 318–329).

Dash, M., & Liu, H. (1997). Feature selection for classification. *International Journal of Intelligent Data Analysis, 1(3),* 131-156.

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

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Stat. Society,39,* 1-38.

Friedman, J.H. (1987). Exploratory projection pursuit. *Journal of American Stat. Association, 82,* 249-266.

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

Jain, A.K., Duin P.W., & Jianchang, M. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22,* 4-37.

Jolliffe, I.T. (1986). *Principal Component Analysis*. Springer Series in Statistics.

Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection. *Proc. of ICML'96* (pp.284-292).

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. *Proc. of ICML-97* (pp. 170-178).

Kowalski, G. (1997). *Information Retrieval Systems Theory and Implementation*. Kluwer Academic Publishers.

Martin, H. C. L., Mario, A. T. F., & Jain, A.K (2002). *Feature Saliency in unsupervised learning*(Technical Report 2002). Michigan State University.

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

Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. *Proc. of SIGIR'00* (pp. 208-215).

Wilbur, J.W., & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of Information Science, 18,* 45-55.

Wyse, N., Dubes, R., & Jain, A.K. (1980). A critical evaluation of intrinsic dimensionality algorithms. *Pattern Recognition in Practice* (pp. 415-425). North-Holland.

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

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

Zamir, O., Etzioni, O., Madani, O., & Karp, R. M. (1997). Fast and Intuitive Clustering of Web Documents. *Proc. of KDD-97* (pp. 287-290).