计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
42(3): 381~386, 2005

一种高效的用于文本聚类的无监督特征选择算法

刘 涛[1] 吴功宜[1] 陈 正[2]
[1](南开大学信息技术科学学院 天津 300071)
[2](微软亚洲研究院 北京 100080)
(liut.netlab@eyou.com)

An Effective Unsupervised Feature Selection Method for Text Clustering

Liu Tao[1], Wu Gongyi[1], and Chen Zheng[2]
[1](*College of Information Technical Science, Nankai University, Tianjin* 300071)
[2](*Microsoft Research Asia, Beijing* 100080)

Abstract Feature selection has been successfully applied to text categorization, but rarely applied to text clustering, because those effective supervised feature selection methods can't be applied to text clustering due to the unavailability of class label information. So a new feature selection method called "*K*-Means based feature selection (KFS)" method is proposed in this paper, which addresses the unavailability of label information by performing effective supervised feature selections on different *K*-Means clustering results. Experimental results show that ① KFS successfully selects out the best small part of features and significantly improves the clustering performance; and ② Compared with other feature selection methods, KFS is very close to the ideal supervised feature selection methods and much better than any unsupervised methods.

Key words feature selection; text clustering

摘 要 特征选择虽然非常成功地应用于文本分类,但却很少用于文本聚类,这是因为那些高效的特征选择方法通常都是有监督的特征选择算法,它们因为需要类信息而无法直接应用于文本聚类.为了能将这些方法应用到文本聚类上,提出了一种新的无监督特征选择算法:基于 *K*-Means 的特征选择算法(KFS).这个算法通过在不同 *K*-Means 聚类结果上使用有监督特征选择的方法,成功地选择出了最为重要的一小部分特征,使文本聚类的性能提高了近 15%.

关键词 特征选择;文本聚类

中图法分类号 TP181

1 引 言

在文本聚类中,文本数据通常采用向量空间模型来进行描述.在这个模型中,每一个单词都作为特征空间坐标系的一维,每一个文本是特征空间中的一个向量[1].这种描述方法简单而且直接,但是同时也使得文本向量空间变得非常的高维而且稀疏.高维稀疏使文本聚类的性能急剧下降,不仅需要花费很长的时间,而且聚类的结果也很难令人满意[2].

为了解决这个问题,最有效的方法就是通过特征选择来进行降维[3,4].特征选择指的是根据一定的规则从原始的特征集中选择一小部分最有效的特征.它根据其规则是否依赖类信息分为有监督的特征选择和无监督的特征选择两类[4,5].

收稿日期:2003-09-26;修回日期:2004-04-29

有监督的特征选择通常通过计算类与特征之间的关系来选择出最具类区分力的特征子集. 这类方法在文本分类上已经得到了非常成功的应用[5,6], 但是因为需要类信息,所以它们无法直接用在文本聚类上. 文本聚类通常使用的是无监督的特征选择. 可无监督的特征选择又因为缺乏类信息而很难选择出最具类区分力的单词,所以当它们应用在文本聚类上时并不是非常有效,不仅不能极大幅度地降低特征空间的维度,而且也很难显著地提高聚类的性能[7,8].

所以,为了能将那些高效的有监督的特征选择应用于文本聚类,本文提出了一种新的无监督特征选择算法——基于 K-Means 的特征选择算法. 这个算法通过合并在不同聚类结果上进行的特征选择来得到最终的特征子集. 实验证明这个算法所得到的聚类结果已经接近理想的有监督特征选择所得到的聚类结果,同时比任何一种无监督特征选择所得到的聚类结果都要优秀得多.

本文接下来还有 4 个部分,其中第 2 节将介绍在文本分类和聚类上最为常用的几种特征选择方法;第 3 节将具体介绍本文新提出的基于 K-Means 的特征选择算法(KFS);第 4 节是实验与分析;最后一部分是总结.

2 各种特征选择算法

在这一节,我们将简单介绍在文本分类和文本聚类应用上最为常用的几种特征选择方法,其中包括两种高效的有监督特征选择方法:信息熵和 χ^2 统计,以及 3 种无监督特征选择算法:文档频数、单词权和单词贡献度[1,5~7].

2.1 信息熵(IG)

信息熵衡量的是某个词的出现与否对判断这个文本是否属于某个类所提供的信息量[5]. 其计算公式如式(1):

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

其中, m 为类的个数, c 代表一个类, t 代表一个单词.

2.2 χ^2 统计(CHI)

χ^2 统计衡量的是一个单词与一个类之间的相关程度,其计算公式如式(2)所示:

$$\chi^2(t) = \sum_{i=1}^{m} p(c_i)\chi^2(t,c_i), \quad (2)$$

其中,

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

N 代表所有单词的总数.

2.3 文档频数(DF)

文档频数是最为简单的一种方法,它指的是在整个数据集中有多少个文本包含这个单词.

2.4 单词权(TS)

单词权的定义如式(3)所示,它计算的是一个词在一对相关文本中的某一个文本中出现的条件下,在另一个文本中出现的概率.

$$TS(t) = p(t \in d_j \mid t \in d_i), d_i, d_j \in D \cap sim(d_i, d_j) > \beta, \quad (3)$$

其中 β 是一个相似阈值,用来判断两个文本是否是相关的文本.

2.5 单词贡献度(TC)

单词贡献度是一种较新的方法[7]. 它认为一个单词的重要性取决于它对整个文本数据集相似性的贡献程度. 其计算公式如式(4)所示:

$$TC(t) = \sum_{i,j \cap i \neq j} f(t,d_i) \times f(t,d_j), \quad (4)$$

其中 $f(t,d)$ 表示的单词 t 在文本 d 中的权重,即"ltc"模式下的 tf * idf 值[9].

3 基于 K-Means 的特征选择算法

特征选择在文本分类的问题上得到了非常成功的应用,尤其是 IG 和 CHI 两种有监督的特征选择方法能够在移走高达 98% 单词的同时还能使分类的精度略有提高[5,6]. 虽是如此,特征选择却很少地应用于文本聚类,因为几乎所有用于文本聚类的无监督特征选择都很难达到令人满意的要求[8]. 它们在不降低文本聚类性能的前提下最多只能移走 90% 左右的单词,随着更多单词的移走,文本聚类的性能会急剧下降[7]. 于是一个很自然的疑问便产生了,那就是如果将在文本分类上成功应用的那些有监督的特征选择用于文本聚类,是否也能大幅度地提高文本聚类的性能? 为了证实这个疑问,Tao 等人做了一组理想实验[7]. 在实验中偷看了标准的类信息,然后使用 IG 和 CHI 这两种有监督的特征选

择方法来进行特征选择. 结果发现有监督的特征选择表现得非常出色,能够在持续提高文本聚类的性能的情况下移走高达98%的单词[7].

但是非常不幸的是,有监督的特征选择无法直接应用于文本聚类,因为这类方法需要依赖类信息,而类信息又正是文本聚类所要解决的问题. 这似乎是个非常尖锐的矛盾,但是也正是这种矛盾给了我们一个新的启发,那就是能否在聚类的结果上使用有监督的特征选择,然后反过来,再在特征选择的基础上进行重新聚类. 这个结果会怎么样? 为了证明这个想法,我们做了一组简单的测试实验,那就是使用 K-Means 聚类算法进行多次不同的聚类,然后在每一个聚类结果上使用 CHI 进行特征选择并再次进行聚类.

非常幸运的是,我们惊喜地发现再次聚类的结果几乎全好于初次聚类的结果,而且当初次聚类的结果越好即与理想的分类结果越接近,在其基础上使用有监督特征选择方法选择出来的特征也和在理想情况下使用相同方法所选择出来的特征越相似,进而再次聚类的结果也就越好.

这个想法很简单,但是在实际情况中却会遇到很大的问题. 一方面理想的类数往往是不可知的;另一方面实际数据的分布往往极不均衡,有的类可能很大,而有的类又可能小到可以作为噪声来处理. 所以在这种情况下,任何一个聚类结果都具有很大的随机性,它们可能把大类打散成很多小类,也很可能把很多小类聚合在一起,而现有的各种方法又很难衡量哪一个聚类结果更接近实际的分类. 因此在单次聚类结果上进行的特征选择也很随机,选得好会提高聚类的性能,选得不好反而会降低聚类的性能.

幸好我们注意到,Fred 和 Jain 为了突破 K-Means 的限制,提出了一种通过合并由不同初始条件而产生的多个不同 K-Means 聚类结果来得到最后的聚类结果的方法[10]. 这个方法有效地消除单次 K-Means 聚类的随机性,能够发现任意形状的簇. 受他们的启发,我们发现,通过合并在不同 K-Means 聚类结果上的特征选择,可以很好地消除在单次聚类结果上进行特征选择的随机性. 因为 K-Means 所得到的每一个解都是一个局部最优解,它其实是从不同的角度刻画了数据的分布规律,所以不同的解之间或多或少都存在一种互补和加强的关系. 正是利用这一点,我们提出了一种新的用于聚类的无监督特征选择算法——基于 K-Means 的特征选择算法,简称 KFS. 这个算法首先通过指定不同的 K 值和初始点来获得不同的 K-Means 聚类结果,然后在不同聚类结果上使用有监督的特征选择来获得在某种程度上互补的特征选择,最后通过合并获得最终的特征选择. 下面就是这个算法的具体描述:

算法 1. 基于 K-Means 的特征选择算法(KFS).

输入:

n,文本的个数;

d,维数,即单词的个数;

K_MIN,最小的聚类个数;

K_MAX,最大的聚类个数;

M,聚类的次数;

FS_Method,特征选择方法,如 IG,CHI.

输出:

$FeatureRanking$,d 维数组,所有单词按照重要性的排序序列.

初始化:

$FeatureRanking$,设置 $FeatureRanking$ 为 d 维的数组且所有元素的重要性值等于 0.

步骤:

1. 循环 M 次

1.1 在[K_MIN, K_MAX]中随机地选择一个 K;

1.2 随机地选择 K 个初始中心点;

1.3 使用前面步骤中得到的参数进行 K-Means 聚类,得到聚类结果 P;

1.4 在 P 的结果上使用 FS_Method 方法为每一个单词求得重要性值;

1.5 更新 $FeatureRanking$ 中每个单词的重要性,即设置 $FeatureRanking(i) = FeatureRanking(i) + FS\_Method(i)$;

2. 对 $FeatureRanking$ 按照重要性进行排序.

从输入输出上看,KFS 算法和所有其他的特征选择方法一样,也是通过某个特殊的规则来计算每一个单词的重要性. 但是与其他算法不同的是,它更类似于一个框架,所有现存的有监督特征选择方法都可以集成到这个框架里作为这个算法内部的核心特征选择算法而存在.

4 实验与讨论

为了更好地评估 KFS 算法,我们采用了多组参数、多种评价标准和多个数据集.

4.1 评价标准

Entropy 和 *Precision* 是两种常用的通过比较聚类结果和标准分类结果来衡量聚类结果好坏的标准.所不同的是, *Entropy* 越小表示聚得越好,而 *Precision* 刚好相反.

4.1.1 *Entropy*

Entropy 衡量的是每一个聚类的纯度[7]. 设 G', G 分别为聚类所得到的簇类的个数和标准类的个数;设 A_i 为聚类结果中的某一个簇类,并设这个簇类中每一个文本 $d_i \in A$, $i=1,\cdots,|A|$ 的标准类标识为 $label(d_i)$,它的值等于标准的类标识 $c_j(j=1,\cdots,G)$. *Entropy* 的计算公式如式(5):

$$Entropy = -\sum_{k=1}^{G'} \frac{|A_k|}{N} \sum_{j=1}^{G} p_{jk} \times \log(p_{jk}), \quad (5)$$

其中, $p_{jk} = \frac{1}{|A_k|} |\{d_i | label(d_i = c_j)\}|$.

4.1.2 *Precision*

Precision 是非常直观的评价标准[7]. 它假设一个簇类的类标识就等于这个簇类中最大的那个标准类的类标识,所以一个簇类的 *Precision* 就等于其中最大的那个标准类所占的比例,其计算公式为

$$Precision(A) = \frac{1}{|A|} \max(|\{d_i | label(d_i) = c_j\}|). \quad (6)$$

然后最终的 *Precision* 是所有簇类 *Precision* 的加权平均:

$$Precision = \sum_{k=1}^{G'} \frac{|A_k|}{N} Precision(A_k). \quad (7)$$

4.2 数据集

为了避免单一数据集可能引起的片面结果,实验使用了两个数据集:Reuters-21578①(Reuters)和网页数据集②(Web). 对于 Reuters 我们只选择那些至少属于一个类并且具有"Lewis split"分割标记的文本,并选择第 1 个类标识作为该文本的标准类标识. 对于 Web,我们随机选择了一个子集. 表 1 显示了它们的基本属性:

Table 1 Datasets

表 1 数据集

Data Sets	Classe Num.	Docs Num.	Terms Num.	Avg. Terms Per Doc.
Reuters	80	10733	18484	40.7
Web	35	5035	56399	131.9

① http://www.daviddlewis.com/resources/testcollections/

② http://dmoz.org/

4.3 实验结果和讨论

实验使用 *K*-Means 作为聚类算法. 因为 *K*-Means 受初始中心点影响较大,所以为了公平起见,我们为每一个数据集都预先生成了 10 组初始中心点. 所以在每一个参数下得到的每一个结果(图中的一个点)都是在这 10 组初始中心点上进行 10 次 *K*-Means 聚类而得到的平均值. 另外,在聚类之前,我们对数据集都做了标准的去停顿词、词根还原和 tf * idf(ltc)加权预处理.

IG 和 CHI 需要类信息,所以在这里将它们作为理想的特征选择,即采用作弊方法使用标准分类信息来进行特征选择. TS 的 β 参数很难自动获得,所以在实验中,我们测试了多个不同的值,然后将其中最好的结果作为 TS 的结果显示出来. KFS 一共有 4 个参数,其中 *FS_Method* 和 *M* 设为 CHI 和 10. *K_MIN* 和 *K_MAX* 对 Reuters 设为 5 和 90,对 Web 则设为 5 和 50.

图 1 和图 2 分别显示了各种特征选择方法在 Reuters 上得到 *Entropy* 结果和 *Precision* 结果. 图 3

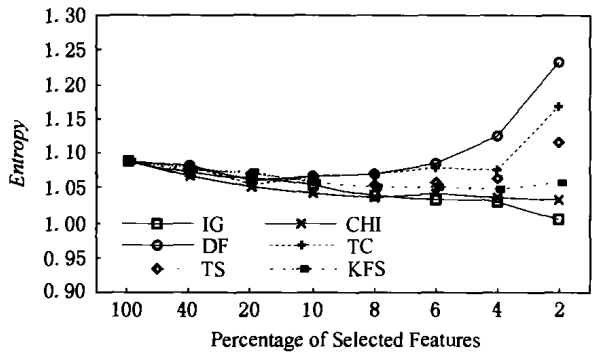

Fig. 1 *Entropy* comparison on Reuters.

图 1 Reuters 上的 *Entropy* 结果

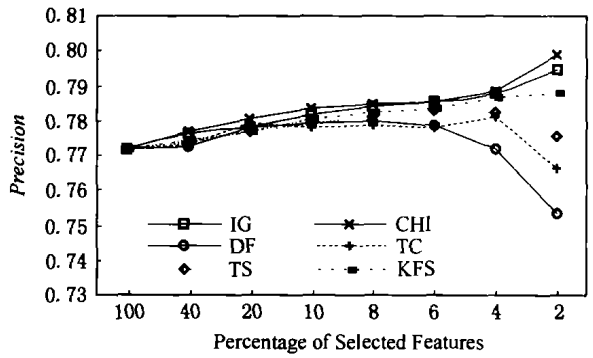

Fig. 2 *Precision* comparison on Reuters.

图 2 Reuters 上的 *Precision* 结果

和图 4 分别显示了各种特征选择方法在 Web 上得到 *Entropy* 结果和 *Precision* 结果. 其中横坐标表示的是选择了百分之多少最为重要的单词,其中 100 表示选择了所有单词,即原始空间,所以在这一点上得到的值便是所有结果的基准值.

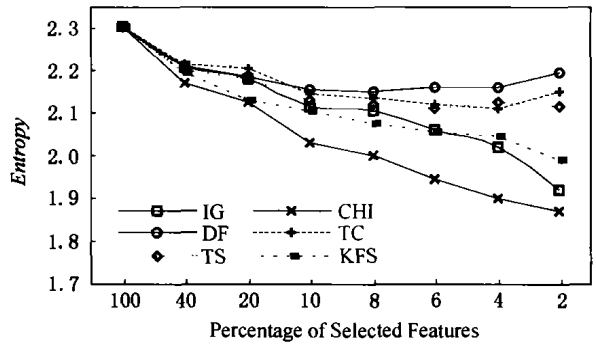

Fig. 3 *Entropy* comparison on Web.

图 3 Web 上的 *Entropy* 结果

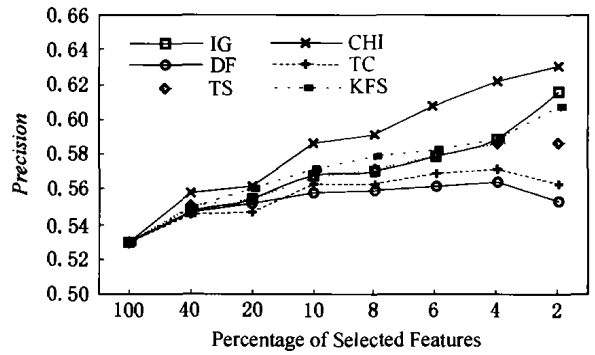

Fig. 4 *Precision* comparison on Web.

图 4 Web 上的 *Precision* 结果

通过这些图,我们能总结出以下几点:

(1) 在两个数据集上,KFS 算法都表现得非常优秀,它在移走高达 98% 单词的时候还能保持聚类性能的持续提高. 例如,在 Web 数据集上,当移走 98% 的单词时,*Entropy* 从基准的 2.305 降到了 1.988(相对 13.8% 的进步),*Precision* 从基准的 52.9% 升到了 60.7%(相对 14.7% 的进步).

(2) KFS 算法所产生的聚类结果已经非常接近 IG 或者 CHI 所产生的结果,这意味着 KFS 虽然是无监督的特征选择方法,但也同样能选择出最为重要的单词.

(3) KFS 算法大大好于任何一种无监督的特征选择方法. 可以看到,当移走少于 90% 单词的时候,所有的曲线都非常接近,但是当移走超过 90% 单词的时候,随着越多单词的移走,这些曲线越来越分开,DF,TS 和 TC 的性能都有不同程度的下降,但是 KFS 却仍能保持性能的持续上升. 例如在 Reuters 上,在移走 98% 的单词后,DF 所产生的 *Entropy* 退步了 13.2%,TC 退步 7.3%,TS 退步 2.4%,但 KFS 仍进步了 2.9%.

(4) 不同的数据集之间的差异很大. Reuters 是人为标定的数据集,文本中没有太多的噪声单词,所以即使最理想的特征选择方法对 Reuters 聚类结果的提高也非常有限. 而 Web 数据集是真实的网页数据集,其中包含了大量的噪声单词,以至于大部分的方法都能较大地提高聚类的性能.

(5) 总体而言,所有的特征选择方法都能移走至少 90% 的单词而不降低聚类的性能. 这也说明了对文本聚类而言,大部分的单词都是噪声或者冗余单词.

在上面的实验中,我们为 KFS 算法直接指定了参数值,这样得到的结果具有一定的片面性. 所以为了更全面地验证 KFS 算法的高效性和鲁棒性,我们在 Reuters 数据集上做了另一个实验,那就是设定不同的 *K_MIN*, *K_MAX* 和 *M* 值来测试 KFS 算法.

我们一共测试了 4 组参数,分别表示为 KFS_1, KFS_2, KFS_3 和 KFS_4. 它们的 *K_MIN*, *K_MAX* 和 *M* 分别取值为 (5,90,10), (5,90,20), (5,40,10), (50,90,10).

图 5 显示的就是在 Reuters 数据集上得到 *Entropy* 结果. *Precision* 的结果图因为和 *Entropy* 类似,所以没显示出来. 从这个图可以看出,所有的曲线都很接近,这也就是意味着 KFS 能较好地平衡参数的影响.

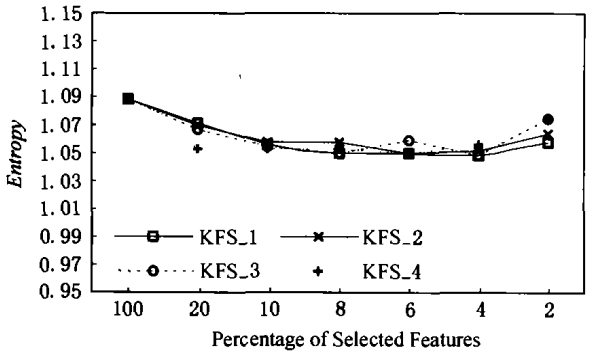

Fig. 5 *Entropy* results on Reuters.

图 5 Reuters 上的 *Entropy* 结果

5 结 论

这篇论文提出了一种新的用于文本聚类的无监督特征选择算法——基于 *K*-Means 的特征选择算

法(KFS). 这个算法最大的贡献在于,通过在聚类结果上使用特征选择的思想,将那些非常有效但无法直接应用到文本聚类上的有监督的特征选择方法成功地应用到文本聚类上,不仅极大地降低了文本数据空间的维度,而且使文本聚类的性能有了显著的提高.

另外,通过将 KFS 算法和其他特征选择方法比较发现,KFS 算法所产生的聚类结果已经非常接近理想的 IG 或者 CHI 所产生的结果,同时大大好于任何一种无监督的特征选择方法.

参 考 文 献

1 Lu Yuchang, Lu Mingyu, Li Fan, *et al*.. Analysis and construction of word weighting function in VSM. Journal of Computer Research & Development, 2002, 39(10): 1205~1210 (in Chinese)
(陆玉昌, 鲁明羽, 李凡, 等. 向量空间法中单词权重函数的分析和构造. 计算机研究与发展, 2002, 39(10): 1205~1210)

2 C. C. Aggrawal, P. S. Yu. Finding generalized projected clusters in high dimensional spaces. The SIGMOD'00, Dallas, 2000

3 M. Dash, H. Liu. Feature selection for clustering. The PAKDD-00, Kyoto, 2000

4 F. Sebastiani. Machine learning in automated text categorization. ACM Computin Surveys, 2002, 34(1): 1~47.

5 Y. Yang, J. O. Pedersen. A comparative study on feature selection in text categorization. The ICML97, Nashville, 1997

6 M. Rogati, Y. Yang. High performance feature selection for text categorization. The CIKM-02, Mclean, 2002

7 L. Tao, L. Shengping, C. Zheng, *et al*.. An evaluation on feature selection for text clustering. The ICML03, Washington, 2003

8 J. G. Dy, C. E. Brodley. Feature subset selection and order identification for unsupervised learning. The ICML'00, Stanford University, 2000

9 G. Salton. Automatic text processing: The transformation, analysis, and retrieval of information by computer. Reading, Pennsylvania: Addison-Wesley, 1989

10 A. Fred, A. K. Jain. Evidence accumulation clustering based on *K*-means algorithm. SSPR/SPR, Windsor, 2002

Liu Tao, born in 1977. Ph. D. His main research interests include text mining, web mining, and user interface.
刘涛,1977 年生,博士,主要研究方向为计算机网络与信息系统、文本挖掘、Web 挖掘、用户界面.

Wu Gongyi, born in 1947. Professor and Ph. D. supervisor. His main research interests include computer network, network security, and Internet technology.
吴功宜,1947 年生,教授,博士生导师,主要研究方向为计算机网络与信息系统、网络与信息安全技术、Internet 技术.

Chen Zheng, born in 1972. Professor of Microsoft Research Asia. His main research interests include machine learning, information retrieval, speech recognition, natural language processing, multimedia information retrieval, personal information management, and artificial intelligence.
陈正,1972 年生,微软亚洲研究院研究员,主要方向为人工智能、神经网络、动态环境下的问题求解.

Research Background

Feature selection has been widely studied on text categorization, but the study seldom touched on text clustering, because the unavailability of class label information makes unsupervised feature selection very difficult. To solve this problem, a thorough research was done and an "iterative feature selection (IF)" method was proposed which addressed the unavailability of label information by utilizing an effective supervised feature selection method to iteratively select features and perform clustering. This research was published on the 20th International Conference on Machine Learning (ICML'03).

It this paper, we extend the above study and propose a new and better feature selection method called "*K*-Means based feature selection (KFS)". Different from IF, KFS performs effective supervised feature selections on different *K*-Means clustering results. Although KFS is much better than any other unsupervised feature selection methods, it still has a big distance to the ideal supervised methods, such as information gain and χ^2 statistic, so the research on unsupervised feature selection is still a big challenge in the text mining field.