基于词语上下文的文本分类研究

杨金柱[1],刘金岭[2]
(1.江南大学 物联网工程学院,江苏 无锡 214122;
2.淮阴工学院 计算机工程学院,江苏 淮安 223003)

摘 要:文本自动分类系统无法直接理解其语义并进行分类,需要对文本进行预处理,提取能表达文本主题内容的关键词,将这些关键词用结构化的形式保存起来,形成文本的表示。针对文本数据中存在大量词语共现的特点,提出了一种基于上下文的文本分类方法。该方法利用词语的上下文关系定义了词语相似度和词语权值,更科学地表达了词语在该类别中的语义表示,从而更能提高文本分类的质量。实验结果表明,该方法的分类效果比传统的简单向量距离分类法有明显的改善。

关键词:词语共现;上下文;词语相似度;文本分类
中图分类号:TP391.1 **文献标识码**:A **文章编号**:1673-629X(2011)08-0145-04

Study of Text Classification Using Context

YANG Jin-zhu[1], LIU Jin-ling[2]
(1. School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China;
2. School of Computer Engineering, Huaiyin Institute of Technology, Huai'an 223003, China)

Abstract: Automatic text categorization system cannot directly understand its semantic and classification, need text pretreatment, extraction can express text topics content keywords, these keywords using structured stored together to form the text representation. According to the common characteristics presented by a large number of words, a context-based text classification method is put forward. This method defines the similarity and weights of words using the context relations between them, which expressed more scientific terms in this category in the semantic representation, thus improve the quality of text categorization better. Experimental results show that the method of classification context-based performance has significantly improved compared with the traditional simple vector distance classification.

Key words: word co-occurrence; context; word similarity; text classification

0 引 言

文本分类是一个根据文本的内容自动确定文本类别的过程。由于文本的结构大都不固定,有些文本并没有结构,甚至有些用自然语言来描述文本的内容,因此所给出的文本分类系统是无法直接对文本集进行分类的,一般地,对文本分类首先需要按语义提取反映文本内容主题的关键词,构成向量,用以表示文本[1]。对大量文本分类的一般过程是:搜集大量文本作为训练集,然后对该训练集进行训练以建立一个分类器,然后对待分类的文本进行特征向量抽取,构成表示该文本的特征向量,再利用该分类器进行文本的类别判断。目前,在文本分类领域,一般是利用向量空间模型(VSM)来表示文本[2],研究表明这是一种较好的表示方法。在文本向量空间模型中,对于每个文本,都已根据文献[3]的方法表示成形如 $SM = < w_1, t_1, w_2, t_2, \cdots, w_i, t_i, \cdots, w_n, t_n >$ 的向量,其中 w_i 是特征词语,t_i 是 w_i 在文本(或类别)中的权值,其含义是 w_i 在文本(或类别)中的重要程度。上述两种特征集合中特征词语的权值有多种计算模式,比较常用的是 TF-IDF 公式。这些权值计算模式大都假设特征词语之间是正交关系,并且假定特征词语的重要性与其在文本中的出现位置无关。如文献[4]在文本分类中利用同义概念归并、上下位概念的聚焦以及文本重点词汇的确定来获取文本的主题,提高了文本的分类速度。文献[5]中利用种子词汇的确定来实现对文本空间的降维,从而提高了文本的分类及簇类主题的提取的速度。文献[6]获取能识别文本类别的特征词语集合是依据训练文本集的特征词环境,该算法忽略了该句子内部特征

收稿日期:2011-01-17;**修回日期**:2011-04-21
基金项目:江苏省淮安市科技计划项目(HAG09061);淮阴工学院重点基金项目(HGA0907)
作者简介:杨金柱(1980-),男,硕士研究生,研究方向为文本数据挖掘;刘金岭,教授,研究方向为数据仓库及文本数据挖掘。

词语之间应该遵循的其他约束,该算法在构造文本分类器时,只关注一个句子由哪些特征词语构成。文献[7]对一些分类贡献较大的特征词语由于其对应的特征值小而被滤掉,尽管该算法能在经过降维的文本向量中仍能包含特征词语之间的潜在语义关系,这样就会影响分类的质量。这些算法都是利用简单的统计方法来确定特征词语的权值,没有考虑到各特征词语之间语义上的相互关系以及这种关系对文本表示质量的影响。这样,很容易会出现文本表示结果与原文本之间的语义差异,近而会影响到文本分类的质量[6]。文中在构建文本分类器和文本表示的过程中,充分考虑到了各特征词语之间语义上的相互关系以及这种关系对文本表示质量的影响,从而提高了分类的质量。

1 特征词语相似度

在上下文处理中,特征词语之间的关系是重要的信息资源[8]。按照上下文理论,正常情况下事物是和某种特定的背景相关联的,这个背景由与该事物有逻辑关系的另外一些事物组成。因此可以认为一个"概念"词的出现总是伴随着与其相关的其他概念的出现[9]。比如,在多篇文章中,词语"计算机"经常与"硬件"、"软件"、"网络"这些词共同使用。而在另外一些含有"电脑"词语的文章中,如果也经常出现"硬件"、"软件"、"网络"这些词,并且它们在文章中出现的位置相近。那么就可以推测"计算机"和"电脑"是两个语义很相近的词。文中是基于这样假设的,在文本集中,两个词语的相似度与这两个词共现于文本的次数和平均上下文距离密切相关。

1.1 两个词语在文本中的平均上下文距离

设 w_i,w_j 是文本向量集 S 中的任意两个词语,它们共现文本中的分布有如下两种情况:一是 w_i,w_j 是共现在同一个文本中;二是 w_i,w_j 没有出现在同一文本中。计算 w_i,w_j 的平均上下文距离(以词为单位)算法如下:

输入:词语 w_i,w_j

输出:平均上下文距离 $d_{avg}(w_i,w_j)$

算法 1:

if 对任意 $SM \in S$,都有 $w_i,w_j \notin SM$ then

$d_{avg}(w_i,w_j)=\infty$;

else

求出 S 中 w_i,w_j 共现文本集合 $\{SM_{p_1},SM_{p_2},\cdots,SM_{p_m}\}$;

$d=0$;

for $i=1$ to m

// m 为 w_i,w_j 在共现文本集 S 中的次数

计算 w_i,w_j 在 SM_{p_i} 中的位置信息 loc(w);

$d=d+|\mathrm{loc}(w_i)-\mathrm{loc}(w_j)|$;

//loc(w_i) 表示词 w_i 在文本中的位置,这里假设两个词语的先后位置不影响它们的相似度

next i

$d_{avg}(w_i,w_j)=d/m$;

end if

输出 $d_{avg}(w_i,w_j)$

1.2 基于上下文的词语语义相似度

设 w_i 和 w_j 是文本向量集 S 中至少共现一次的两个词语(本部分对在 S 中 w_i 和 w_j 没有共现于一个文本的情况不予考虑)。

这里取计算 w_i 和 w_j 的上下文关系的公式如下:

$$C(w_i,w_j)=tf(w_i,w_j)\times\log\frac{|S|}{|<w_i,w_j>|} \tag{1}$$

其中,$|S|$ 表示文本向量集 S 中所包含的词的个数,$|<w_i,w_j>|$ 表示 S 中 w_i 和 w_j 至少共现一次的词的数量。

$$tf(w_i,w_j)=\begin{cases}1+\log n(w_i,w_j), & \text{if } n(w_i,w_j)>0\\ 0, & \text{其它}\end{cases} \tag{2}$$

其中,$n(w_i,w_j)$ 表示 SM 中包含这两个词的文本数。

下面考虑 w_i 和 w_j 的相互贡献,文中称为两个词语的相关度,记为 rela(w_i,w_j),这里有:

$$\mathrm{rela}(w_i,w_j)=\frac{C(w_i,w_j)}{\sqrt{\sum_{s=1}^{|S|}C(w_i,w_j)}} \tag{3}$$

rela(w_i,w_j) 的意义在于,如果 w_i 与 w_j 同时出现在 S 中文本的数量越多,说明 w_i 与 w_j 之间刻画的上下文关系越重要,换言之,词语 w_i 和词语 w_j 之间的相互贡献就越大。另一方面,如果 S 中与 w_i 或 w_j 共现的不同词越多,说明 w_i 与 w_j 相互贡献就越小,分母开方的目的也说明了影响程度相对要小。

最后定义 w_i 和 w_j 的相似度。由于在一个文本中随着句子之间的距离增大,两个词之间的关联关系就会表现的越来越弱。如果在同一个文本中,用 $d_{avg}(w_i,w_j)$ 表示词 w_i 与词 w_j 之间的平均上下文距离。文中取词语 w_i 和 w_j 的基于上下文语义相似度 sim(w_i,w_j) 为文献[10]中公式(6):

$$\mathrm{sim}(w_i,w_j)=\mathrm{rela}(w_i,w_j)\times\frac{e^{-\lambda d_{avg}(w_i,w_j)}}{\sum_{w,w_i,w_j\in SM}e^{-\lambda[d_{avg}(w,w_i)+d_{avg}(w_j,w)]}} \tag{4}$$

其中 λ 是影响因子。

2 基于上下文关系的特征词语权值

特征权值计算模式有多种,TF-IDF 公式是比较常

用的。该计算模式只是考虑到词语在文本中出现的频数,既没有考虑到词语在文本中的位置又没有体现出词语在整个文本集中的状况,这些与该词语反映文本的内容及文本与类别的关系是非常重要的。文中将进一步修改由文献[1]中定义的词语权值,以便更好地反映文本的语义类别。

2.1 基于上下文关系的同义词

如果两个词分别与多个词语共现在多个文本中,而且它们在文本中出现的位置也基本上是相同的,那么将该两个词定义为上下文同义词,如前面谈到的两个词语"计算机"与"电脑"。定义如下:

定义 1 设 w 是文本 SM 中的任意一个词语,$Q=\{w_1,w_2,\cdots,w_s\}$ 是文本集 S 中至少与 w 共现一次的词语序列。定义 w 关于词语序列 Q 的相似度为:

$$R(Q,w)=\sum_{i=1}^{s}\mathrm{sim}(w_i,w) \tag{5}$$

定义 2 设 w_i 和 w_j 是文本集 S 中的两个词语,$\varepsilon>0$ 是预先给定的阈值,如果

$$|R(Q,w_i)-R(Q,w_j)|<\varepsilon \tag{6}$$

则称 w_i 和 w_j 关于 S 是基于上下文同义词。

定义文本集 S 上下文同义词的意义在于利用文献[2,3]中同义词概念归并、上下文概念聚焦进行文本分类,起到降维的作用以提高分类的速度。

2.2 基于上下文关系的词语权值

文本分类的过程是在文献[4,5]中的文本向量集 S 的基础上进行同义词语义归并和权值定义,再利用下面的方法进行词语上下文权值定义,其意义在于使词语更能体现在文本分类类别中的重要程度。

定义 3 假设 $w_i\in SM$,t_i 是经过定义 2 中上下文同义词归并后得到的权值。给定一个定数 m_0,$QE=\{w_{i_1},w_{i_2},\cdots,w_{i_n}\mid w_{i_k}\in S$ 但 $w_{i_k}\notin SM\}$ 是与 w_i 至少共现 m_0 次以上的词语,定义 w_i 的基于上下文词语权值为:

$$\mathrm{cont}_i=t_i+\sum_{k=1}^{m}\mathrm{sim}(w_i,w_{i_k}) \tag{7}$$

定义 3 中词语 w_i 的上下文权值的意义是这样考虑的,一方面,t_i 反映了 w_i 对 SM 的贡献,另一方面,由于考虑的是分类,所以还应考虑到 w_i 对于类别的贡献。

3 基于上下文的文本分类

文中的主要目的是探讨词语上下文关系对文本自动分类效果的影响,而这种影响主要体现在特征选择、词语相似度、同义词表示和权值的表示等与文本处理有关的环节中,没有涉及到文本的具体的分类算法,为此,文中的方法将侧重点放在利用词语上下文关系来改善待分类文本的表示质量上。分类器设计的思想是对测试文本作类别测试,根据选定的大容量文本训练集,手工进行分词,利用公式(6)定义上下文语义相似度,对于给定的阈值找出训练集中的上下文同义词,构成上下文同义词库,并利用文献[5],进行同义词概念归并和上下位概念聚焦,从而达到降维的效果,再利用公式(7),进行上下文权值的定义,利用文献[4]的方法构造分类器。在被测试文本的处理上,先对测试文本做前面的上下文预处理(利用到训练文本中的上下文同义词库),得到测试文本的类别特征向量,利用分类器进行类别鉴定。分类模型如图 1 所示。

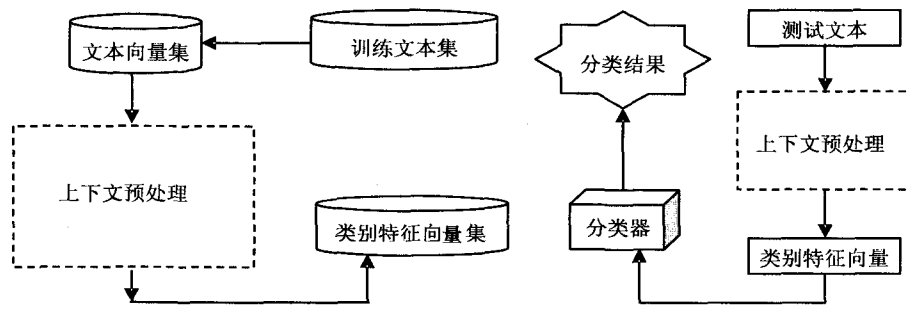

图 1　基于词语上下文的文本分类模型

4 实验结果及分析

试验算法用 Visual Basic6.0 和 Access2000 实现,在内存为 2.0GB,主频为酷睿双核 2.0GHz,操作系统为 Windows XP 的方正计算机上进行实验。实验语料选自于复旦大学李荣陆博士收集整理的中文文本分类语料库[11](包括训练集和测试集),该语料库适于小规模研究。去掉该语料库中的重复文档后,在 8214 篇训练文档中选取了 5 个较大的类,分别为:政治类、教育类、经济类、军事类和环境类。其中政治类的 400 篇,教育类的 200 篇,经济类的 200 篇,军事类的 100 篇,环境类的 100 篇。并有意识地在每一类中加入了约 5% 的噪声文本数据,构成文档总数为 1000 篇的语料。为了试验节省时间,首先进行手工预处理:将这 1000 条中文文本组合成 1000 条记录的 N 维的 VSM 空间录入到数据库中。

为了试验叙述的方便,将文献[4]中的基于主题的中文文本分类算法记为 Theme_ classification,将文献[5]中的基于降维的文本语义分类及主题提取算法仍记为 Coarse_classification,文中基于上下文的文本分类算法记为 Context_ classification。

4.1 分类质量对比实验

对文本进行分类常用的质量评估标准有分类查准率 P(Precision) 与查全率 R(Recall),查准率与查全

率的几何平均数,信息估值(Information Score),兴趣度(Interestingness)等[12],其中查准率 P 是所判断的文本与人工分类文本吻合的文本所占的比率;查全率 R 是人工分类结果应有的文本与分类系统吻合的文本所占的比率。而查准率和查全率反映了分类质量的两个不同方面,两者必须综合考虑,不能偏废。因此,可以综合考虑查准率和查全率的一个新的评估指标测试值 $F1$。文中采用评估文本分类的主要三个指标:查准率 P、查全率 R 和测试值 $F1$,公式分别为:

$$P = \frac{\text{分类的正确文本数}}{\text{实际分类的文本数}} \times 100\% \quad (8)$$

$$R = \frac{\text{分类的正确文本数}}{\text{应有的文本数}} \times 100\% \quad (9)$$

$$F1 = \frac{\text{查准率} \times \text{查全率} \times 2}{\text{查准率} + \text{查全率}} \quad (10)$$

由于词语共现能反映文本类别的特征,Context_ classification 分类方法利用词语的共现的上下文关系定义了词语相似度、同义词和权值的概念,进一步增加了词语的语义类别特征;而 Theme_classification 分类方法是从定义《知网》"义原"的相似度出发,进而又定义了词语相似度、文本的相似度,分类质量还是比较高的。而在 Coarse_classification 分类方法中由于选取了种子词汇来达到降维的目的,从而降低了分类质量。实验主要评估参数值比较结果如图 2 所示。该试验中所取的 Coarse_classification 种子词汇的概率为 0.2,公式(6)中 ε 取 0.01。

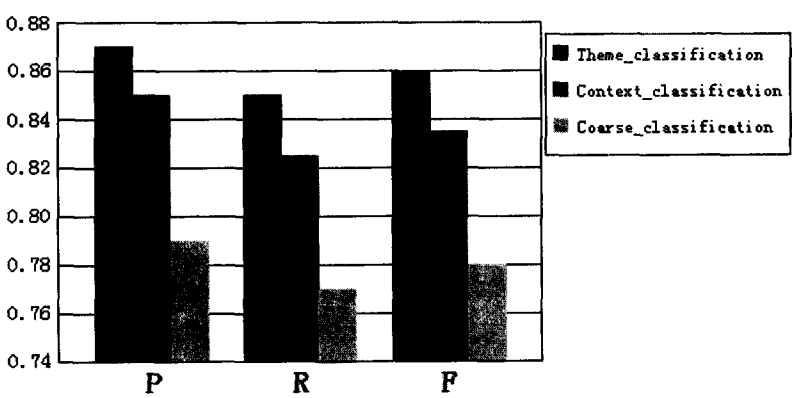

图 2 三种分类实验评估参数值比较

从图 2 可以看出,Context_classification 分类质量略低于 Theme_classification 但远高于 Coarse_classification。笔者又分别对 5 大类训练集增加了 50% 的文本记录进行试验表明,随着训练集文本数量的增大,Context_classification 分类的质量进一步提高。

4.2 分类时间对比实验

由于 Context_classification 采用了文本词语共现方法定义了词语相似度,并重新定义了词语的上下文同义词(降维)和上下文权值,这些通过一次扫描文本向量集就能完成,因而比 Theme_classification 和 Coarse_classification 分类方法中都利用文献[1]中通过《知网》义原树进行相似度定义的多次的多重循环嵌套节省大量的时间。但在 Coarse_classification 分类方法中是通过选取种子词汇来达到降维的目的,因此以牺牲文本的分类质量为代价而提高了文本分类的速度。实验分别对 200、400、600、800、1000 条文本进行了实验,分类耗时结果如图 3 所示。

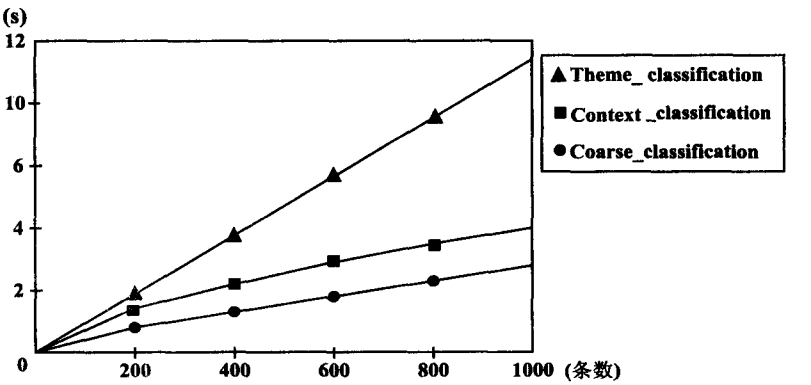

图 3 三种分类实验运行时间比较

从图 3 可以看出 Context_classification 分类耗费时间略高于 Coarse_classification 分类,但远低于 Theme_classification 分类耗费的时间。进一步实验表明,随着训练集文本数量的增大,Context_classification 分类与 Coarse_classification 分类耗费时间更接近。

5 结束语

针对文本集容量大的特点,文中给出了基于词语上下文关系的文本的语义分类方法,改进了传统的简单向量距离分类法。主要利用词语的上下文关系定义了词语上下文语义相似度和上下文权值,充分考虑了词语在文本中的上下文关系。实验结果表明,对于海量的文本数据,该方法在综合考虑分类耗时和分类质量时效果非常明显。

参考文献:

[1] 郭少友.一种基于词上下文向量的文本自动分类方法[J].情报科学,2008,26(7):1030-1034.

[2] Shafiei M. A systematic study of document representation and dimension reduction for text clustering[EB/OL]. 2007-05-03. http://PPwww. cs. dal. ca/research/techreports/2006/CS-2006-05. shtml.

(下转第 152 页)

linux 版本，Hadoop 软件版本为 Hadoop-0.21。

实验的输入数据是大批量的 XML 文件，利用 Hadoop 加载不同的 MapReduce 模型对 XML 文件进行数据提取，即从这些文件中查找指定的内容。通过对比连续执行、并行传统模型、并行新模型三种方法在上级节点域为 200 的情况下的执行效率，经统计在上级节点域固定的情况下，改进后的模型，树节点域 20 比 10 的效率明显提高了 11% 左右。从图 6 可以看出，随着问题的规模的增加，改进后的模型效率得到了提高，传统模型随着规模的增加其运行时间陡然上升，而改进后的模型其运行时间趋于稳定。经过计算，改进后的模型效率比传统模型提高了 13.8%。

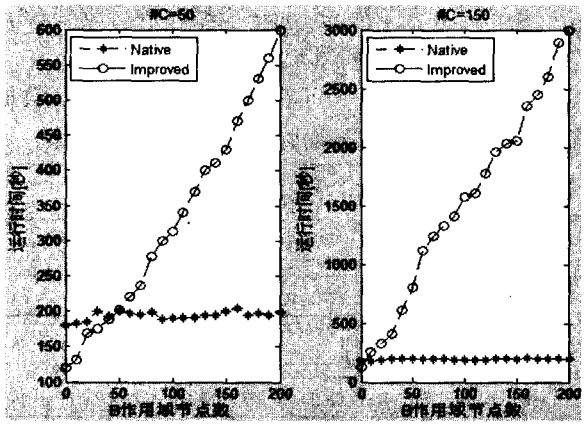

图 6 效率趋势图

4 结束语

文中为使 MapReduce 模型更具有一般性，特提出一种新的聚类聚合模型，它将所有数据抽象构造成树，形式化关联信息载入 MapReduce 模型。另在 Reduce 函数阶段，利用< $k1$, $k2$, …, kn, value > 代替 < k, value>。通过实验分析发现，改进模型的效率明显得到提高，达到了预期的目的。

当然文中也存在很多不足，新模型的提出，忽略了 MapReduce各个任务切换调度时所花费的时间，通过实验看到，该部分的花销应该适当加以考虑，建立更加完善的模型。另外，由于实验条件限制，并没有在高并发、大容量、高压力下的环境下考虑模型的稳定性。

参考文献：

[1] Dean J, Ghemawat S. MapReduce: Simplied Data Processing on Large Clusters[C]// Proceedings of the 6th Conference on Symposium on Operating Systems. Design & Implementation. [s. l.]: USENIX Association, 2004.

[2] Catanzaro B C, Sundaram N, Keutzer K. A Map Reduce Framework for Programming Graphics Processors[C]// Workshop on Software Tools for MultiCore. [s. l.]: [s. n.], 2006.

[3] Ranger C, Raghuraman R, Penmetsa A, et al. Evaluating MapReduce for Multi-core and Multi-processor Systems[C]// HPCA. [s. l.]: [s. n.], 2007: 13-24.

[4] 郑启龙，房 明，汪 胜，等. 基于 MapReduce 模型的并行科学计算[J]. 微电子学与计算机，2009，26(8)：21-23.

[5] Sarje A, Aluru S. A MapReduce Style Framework for Trees [R]. [s. l.]: Department of Electrical and Computer Engineering, 2008: 17-18.

[6] 胡 彧，封 俊. Hadoop 下的分布式搜索引擎[J]. 计算机系统应用，2010，19(7)：24-26.

[7] 焦金涛. 基于 PageRank 的 Web 挖掘改进算法[J]. 计算机工程，2009，35(15)：31-32.

[8] 史佩昌，王怀民. 面向云计算的网络化平台研究与实现[J]. 计算机工程与科学，2009，31(11)：12-13.

[9] 孙广中，肖 峰. MapReduce 模型的调度及容错机制研究[J]. 微电子学与计算机，2007，24(9)：37-38.

[10] 奚建清，游进国. 基于 MapReduce 的封闭立方体并行计算方法[J]. 华南理工大学学报，2009，37(1)：8-9.

[11] Hadoop. The Apache Software Foundation[EB/OL]. 2010. http://hadoop. apache. org/core.

[12] Bialecki A, Cafarella M, Cutting D, et al. Hadoop: a framework for running applications on large clusters built of commodity hardware [EB/OL]. 2005. Wiki at http://lucene. apache. org/hadoop.

(上接第 148 页)

[3] 刘金岭. 基于语义的高质量中文文本聚类算法[J]. 计算机工程，2009，35(10)：201-205.

[4] 刘金岭. 基于主题的中文短信文本分类研究[J]. 计算机工程，2010，36(4)：30-32.

[5] 刘金岭. 基于降维的短信文本语义分类及主题提取[J]. 计算机工程与应用，2010，46(23)：159-161.

[6] 丘志宏，宫雷光. 利用上下文提高文本聚类的效果[J]. 中文信息学报，2007，21(6)：109-115.

[7] Tombros A. Reflecting user information needs through query based summaries[R]. Glasgow: Department of Computing Science, University of Glasgow, 1997.

[8] 曾雪强，王明文，陈素芬. 一种基于潜在语义结构的文本分类模型[J]. 华南理工大学学报(自然科学版)，2004，32(z1)：99-102.

[9] Gong Leiguang. Exploring Computational Mechanism for Contexts[J]. IEEE Computational Intelligence Bulletin, 2002, 1(1): 19-25.

[10] 刘金岭. 基于查询词扩展的中文垃圾检索研究[J]. 计算机工程，2011，37(3)：151-154.

[11] 李荣陆，王建会，陈晓云，等. 使用最大熵模型进行中文文本分类[J]. 计算机研究与发展，2005，42(1)：94-101.

[12] 徐长青. 中文文本分类技术研究[D]. 长春：吉林大学，2007.