软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(6):1578−1587 [doi: 10.3724/SP.J.1001.2012.04111]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于增量型聚类的自动话题检测研究*

张小明, 李舟军+, 巢文涵

(北京航空航天大学 计算机科学与工程系,北京 100191)

Research of Automatic Topic Detection Based on Incremental Clustering

ZHANG Xiao-Ming, LI Zhou-Jun+, CHAO Wen-Han

(Department of Computer Science and Engineering, BeiHang University, Beijing 100191, China)

+ Corresponding author: E-mail: lizj@buaa.edu.cn

Zhang XM, Li ZJ, Chao WH. Research of automatic topic detection based on incremental clustering. *Journal of Software*, 2012,23(6):1578−1587. http://www.jos.org.cn/1000-9825/4111.htm

Abstract: With the exponential growth of information on the Internet, it has become increasingly difficult to find and organize relevant material. Topic detection and tracking (TDT) is a research area addressing this problem. As one of the basic tasks of TDT, topic detection is the problem of grouping all stories, based on the topics they discuss. This paper proposes a new topic detection method (TPIC) based on an incremental clustering algorithm. The proposed topic detection strives to achieve a high accuracy and the capability of estimating the true number of topics in the document corpus. Term reweighing algorithm is used to accurately and efficiently cluster the given document corpus, and a self-refinement process of discriminative feature identification is proposed to improve the performance of clustering. Furthermore, topics' "aging" nature is used to precluster stories, and Bayesian information criterion (BIC) is used to estimate the true number of topics. Experimental results on linguistic data consortium (LDC) datasets TDT-4 show that the proposed model can improve both efficiency and accuracy, compared to other models.

Key words: topic detection and tracking; TDT; topic detection; incremental clustering; reweighting

摘 要: 随着网络信息飞速的发展,收集并组织相关信息变得越来越困难.话题检测与跟踪(topic detection and tracking,简称 TDT)就是为解决该问题而提出来的研究方向.话题检测是 TDT 中重要的研究任务之一,其主要研究内容是把讨论相同话题的故事聚类到一起.虽然话题检测已经有了多年的研究,但面对日益变化的网络信息,它具有了更大的挑战性.提出了一种基于增量型聚类的和自动话题检测方法,该方法旨在提高话题检测的效率,并且能够自动检测出文本库中话题的数量.采用改进的权重算法计算特征的权重,通过自适应地提炼具有较强的主题辨别能力的文本特征来提高文档聚类的准确率,并且在聚类过程中利用 BIC 来判断话题类别的数目,同时利用话题的延续性特征来预聚类文档,并以此提高话题检测的速度.基于 TDT-4 语料库的实验结果表明,该方法能够大幅度提高话题检测的效率和准确率.

关键词: 话题检测与跟踪;TDT;话题检测;增量型聚类;权重计算

* 基金项目: 国家自然科学基金(61170189, 61003111); 国家教育部博士点基金(20101102120016); 国家重点实验室基金(SKLSDE-2011ZX-03)

收稿时间: 2009-08-07; 定稿时间: 2011-09-01

中图法分类号: TP391 文献标识码: A

随着近年计算机技术的快速发展,网络上的信息已经呈现爆炸式的增长趋势.如此大量的信息个人是无法对它进行有效的组织管理,尤其对于决策者来说,面对如此大的信息量,如果没有有力的工具支持,很难在正确的时间做出正确的判断,因为要决策者去监控和发现所有有关的信息是不现实的.因此,迫切需要一个系统自动发现和组织相关的信息,并且以用户可读的方式展现出来.

话题检测与跟踪(TDT)的目标就是检测相关信息并跟踪事件的发展变化,TDT 的研究成果可以应用到国民经济的多个领域,包括新闻报道追踪、金融股票市场分析、大规模网络信息自动处理等.而话题检测是 TDT 的一个重要的基本任务,它的研究内容主要是在多个新闻流中把讨论相同话题的新闻文档聚类到一起[1−3].话题检测分为回顾式的话题检测和在线的话题检测两个研究方向,回顾式的话题检测即检测已有新闻库中尚未发现的话题,在线的话题检测即在线的检测当前到达的新闻所属的话题.不管那种形式的话题检测都缺乏话题的先验知识,由于要把报道相似内容的无标签新闻组织到一块,因此,聚类算法是解决话题检测的一个有效方法[1],本文主要研究回顾式的话题检测.

但是话题检测又不等同于简单的聚类算法,一个重要的不同之处在于,话题检测研究的对象具有时间性,它们都有发生的先后顺序.另外,话题都只持续一段时间,随后消失或报道减少.而聚类算法是利用类别信息把话题内容相关的文档聚类到一个类中.在本文中,我们定义话题为讨论一致的话题或概念的新闻集,两篇讨论不同特定问题的新闻文档可以属于同一个话题,例如,一篇报道森林火灾的文章和一篇报道地震的文章都属于自然灾害这个话题.对于话题检测,我们的目标在于把叙述一个话题的产生与发展的所有新闻文档聚到一个代表该话题的类中.

影响话题检测性能的另一个重要方面就是对话题相似性的判断,由于每个话题都可能包含不同的叙述方面,因此如果只用一个中心向量来表示一个话题,是很难全面地概括该话题的.在本文中,我们把每个话题用若干子话题来表示,话题之间的相似性由最相似的子话题之间的相似值决定.相似性判断需要注意的另一个方面是:有些特征词在某些话题出现比较频繁,而在其他话题中出现次数很少;而又有些特征词在整个文档库中出现比较频繁.因此,不同特征词对于区分不同话题的作用是不一样的.也就是说,每个话题都有基于该话题的特征词,这些特征词对于区分该话题比较重要.有些文章利用话题的名词实体来提高话题辨别能力,但是这些方法对话题检测性能的提高是有限的[4,5].在本文中,我们利用增量型的方法在话题检测过程中不断提炼基于话题的特征词,给予这些特征词更大的权重,从而提高话题区分能力.

本文提出了一个基于增量和自动聚类算法的话题检测方法,每个话题由多个子话题组成,话题之间的相似性由两个话题中最相似的两个子话题的相似值决定,在聚类的过程中不断提炼基于话题的高辨别性特征词向量,利用该向量表示话题,从而提高话题检测的准确率,通过计算 BIC(Bayesian information criterion)来判断两个类能否被合并,从而自动地决定新闻库中话题的数量.实验结果表明,该话题检测方法相比同类方法具有更高的准确率和效率,并且能够以比较小的错误率来估计话题的数量.

1 相关工作

目前,已经有一些话题检测方面的工作,其中一部分的研究工作是利用语言模型或概率模型的方法.例如,有些文献[6−11]利用了一个多阶段的产生步骤,其中,像作者和标题这样的语义模式被提取出来作为中间步骤,然后利用这些模式的多项式分布来产生词语.这些工作对产生过程的设计各不相同:文献[7]利用基于话题分布的文档产生过程;文献[8]同时产生词语和词语的时间标签两个部分,从而检测话题随时间的变化.其他一些文献则把语言模型扩展到利用链接信息,例如,文献[10]把对文档的引用列表作为该文档“词袋(bag of words)”的另外一个特征.文献[9]在产生过程中引入话题相关矩阵来克服 Dirichlet 分布在描述话题相关性方面的不足.文献[11]利用 EM 算法建立混合模型,进而得到文档集中的话题.

另外一些研究工作利用了分类和聚类的思想.文献[12]利用了扩展的领域信息来建立类,目的是使该类能

够更好地描述话题.该文还表明,对于从多个渠道获得的新闻,利用该方法建立话题具有很优的性能.文献[1]提出了改进的聚类算法,并且与现有的聚类算法进行了比较,基于该聚类算法的话题检测方法利用了神经网络的思想,在建立话题的过程当中,当新的实例和当前类之间出现本质的不同时建立一个新的类,通过调整权重向量来决定类之间的运动,利用该方法它试图找到最优的话题数量.文献[4]提出了一个基于双特征向量的话题检测方法,每个话题利用名字实体和关键词向量来表示,在话题检测的过程中不断调整话题的中心.

然而,这些方法很少利用话题的时间段特性,即每个话题的出现和消失都发生在一段时间内,而且每个话题可能包含子话题,这些子话题可能讨论话题的不同方面,有些子话题之间可能差异比较大,如果只用一个中心向量是不能很好地表示话题的.另外,话题数量的自动判断仍然需要深入的研究.

2 基于增量聚类的自动话题检测

本文采用一种基于增量聚类算法的话题检测算法,该算法利用话题的时间特性,通过不断提炼能够代表话题的特征来提高聚类的性能.该聚类算法能够更加准确和高效地把讨论同一话题的新闻聚到一个话题或者类中,并且能够自动地检测话题的数量.

2.1 特征权重计算

当前,许多的研究工作都利用 TF-IDF 模型来对特征的权重进行计算,并且已经取得了较好的效果,TF-IDF 模型的基本思想是:特征在越少的文档中出现,则该特征的权重越大;或者在文档中出现的次数越多,则该特征的权重越大.但是,TF-IDF 模型的权重计算不能很好地反映以下情况:

(1) 在某一类文档中该特征出现比较频繁,而在其他文档中出现比较少;

(2) 包含某个特征的文档虽然不是很多,但是这些文档比较分散,即分散在各个话题当中.

因此,我们引入一个定义,即特征的话题辨别能力.它表示一个特征对于区分话题的能力,当一个特征在一个话题出现比较频繁而在其他话题出现比较稀少的情况下,该特征具有较高的话题辨别能力.显然:在上面第 1 种情况下,该特征应具有高的话题辨别能力;而第 2 种情况下,该特征应赋予较少的权重,因为它在各个话题中都出现,对区分话题的作用不是很大.为克服这个缺陷,在下一节中,我们采用基于最大熵的方法对权重计算方法及特征的选择进行改进.

另外,TF-IDF 模型仅考虑文档特征的分布,没有考虑文档本身的分布对特征权重的影响.实际上,文档的分布对特征的重要性也是有很大影响的.这个观点主要是基于以下的考虑:文档越重要,则对特征权重的贡献越大,否则更小.基于这个观点,我们采用如下公式计算特征的权重:

$$W(i,j)=LT(i,j)\times GT(i)\times GD(j)\times KD(i) \tag{1}$$

在本文将要用到的符号含义如下:

- tf_{ij}:文档 j 中特征 i 出现的频率;
- dl_j:文档 j 的长度;
- df_i:包含特征 i 的文档的个数;
- df_{ci}:在类 c 中包含特征 i 的文档的个数;
- gf_i:在文档集中特征 i 总共出现的次数;
- sgf:文档集中所有特征出现次数的总和;
- N_c:类 c 中文档的数量;
- N_t:文档集中包含的文档的数量.

$KD(i)$用来衡量特征 i 的话题辨别能力,将在下一节介绍.我们把 TF-IDF 中的 TF 改为如下计算公式,这个公式考虑了文档的长度,因为有些文档比较长,相应的特征在该文档出现的次数会比在短文档出现的次数多.

$$LT(i,j)=\frac{\log(tf_{ij}+1)}{\log dl_j+1} \tag{2}$$

最大熵理论被用到 $GT(i)$和 $GD(j)$中,$GD(j)$主要用来计算文档分布对特征权重的贡献,而 IDF 被 $GT(i)$所替换,它考虑了特征在不同文档的分布情况:

$$GT(i)=1-\frac{H(d\mid t_i)}{H(d)} \tag{3}$$

$$H(d\mid t_i)=-\sum p(j\mid i)\log p(j\mid i) \tag{4}$$

$$p(j\mid i)=\frac{tf_{ij}}{gf_i} \tag{5}$$

$$H(d)=\log N_t \tag{6}$$

$$GD(j)=1-\frac{H(t\mid d_j)}{H(t)} \tag{7}$$

$$H(t)=-\sum p(t)\times\log p(t)=-\sum_{i=1}^{n}\frac{gf_i}{sgf}\times\log\frac{gf_i}{sgf} \tag{8}$$

$$H(t\mid d_j)=-\sum p(i\mid j)\log p(i\mid j) \tag{9}$$

$$p(i\mid j)=\frac{tf_{ij}}{dl_j} \tag{10}$$

2.2 话题特征的提炼

在许多的聚类算法中,所有的特征都平等的对待,并且特征的选择没有利用话题本身的信息.当一个文档的特征向量由许多无话题辨别能力的特征组成时,该文档可能会被聚到一个错误的类中.为了提高聚类的准确率,我们从初始的聚类结果中得到一组具有话题辨别能力的特征,然后在聚类过程中逐步修正提炼这些特征.这样可以减少具有较低话题辨别能力的特征影响,提高话题检测的准确率.本文采用以下公式来衡量一个特征的话题辨别能力的大小:

$$KD(i)=KL(P_{ci}\|P_{ti}) \tag{11}$$

$$P_{ci}=\frac{df_{ci}}{N_c},P_{ti}=\frac{df_i}{N_t} \tag{12}$$

$$KL(P\|Q)=\sum p(x)\log\frac{p(x)}{q(x)} \tag{13}$$

上述式子中,P_{ci}表示在一个类 c 中包含特征 i 的文档数与类中文档数的比值,P_{ti}表示所有包含该特征的文档数与文档集大小的比值.

显然,$KD(i)$对那些出现在某一个话题中次数比较多而在其他话题中出现次数少的特征赋予较大的值,而对话题辨别能力小的特征赋予较少的值.因此,$KD(i)$值能够较好地反映一个特征的话题辨别能力.

在本文的实现中,我们保留那些 KD 值大于一个阈值 T 的特征,在进行特征的权重计算时再考虑特征的 KD 值(见式(14)).同时,在话题聚类的过程中不断筛选调整特征向量,具体实现如图 1 所示的第 9 行伪代码,提炼后,特征权重计算如下:

$$W(i)=W(i)(1+KD(i)) \tag{14}$$

```
Procedure Detection(S,c)
1.    {
2.    ST=build_track(S);
3.    N_d=heap_pair(ST)
4.    while(N_d!=∅){
5.    X,Y=extract_closest_pair(N_d);
6.    Z=merge(X,Y,c);
7.    if (BIC(Z)>BIC(X,Y))){
8.    update ST;
9.    refine feature set;
10.   }}}
```

Fig.1 Procedure of topic detection

图 1 话题检测过程伪代码

2.3 BIC值

在话题检测方法研究当中,怎样自动地估计新闻库中话题的数量是一个开放的问题,也就是自动估计类的个数.有些聚类算法采用密度变化判别的方法来决定类的合并,从而自动地得到类的个数.但是密度变化判别方法需要引入密度阈值,不利于自动地检测话题数量.而在其他聚类算法当中,大都采用模型选择的方法来得到类

的个数,该方法不需要用户输入阈值参数.例如,作为对 *k*-means 聚类算法的扩张,*x*-means 算法[13]采用 BIC 值估计文档集中类的个数.实验证明,BIC 能够以较少的误差估计文档集中类的个数.本文在话题检测的过程中,通过计算 BIC 值来估计话题的个数.

模型选择的主要问题就是从众多的候选模型中选择一个最优的模型,假设$\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$为一个输入数据集,其中的$\boldsymbol{x}_i\in R^d$,$D$ 被划分为 k 个子集 $C_1,\ldots,C_k$,其中,任两个子集都不相交,模型 M_i 的 BIC 值为如下定义:

$$BIC(M_i)=\hat{l}_i(D)-\frac{p_i}{2}\times\log n \tag{15}$$

其中,$\hat{l}_i(D)$ 是在模型 M_i 中数据取最大概率时的 log 值,p_i 是模型 M_i 中独立参数的个数.可以看出,BIC 包含两个部分:第 1 项评价了参数化的模型,预测了这些数据的优劣程度;第 2 项对模型的复杂度进行一个惩罚[4].

一个数据 $\boldsymbol{x}_i$ 属于类 C_j 的概率可以由两部分的概率相乘得到:一部分是 C_j 出现的概率,另一部分是 x_i 规范化的多项式密度值.因此,可以得到如下计算数据 $\boldsymbol{x}_i$ 属于类 C_j 的概率值:

$$\hat{P}(x_i)=\frac{n_j}{n}\cdot\frac{1}{\sqrt{2\pi}\hat{\sigma}^d}\exp\left(-\frac{1}{2\hat{\sigma}^2}\|x_i-\mu_j\|^2\right) \tag{16}$$

其中:n_j 是类 C_j 中元素的个数;$\hat{\sigma}^2$ 是方差的最大概率值,由以下公式给出:

$$\hat{\sigma}^2=\frac{1}{n-k}\sum_i(x_i-\mu_j)^2 \tag{17}$$

因此,C_j 中数据的最大概率的 log 值由以下式(18)计算:

$$\begin{aligned}\hat{l}(C_j)&=\log\prod_{i\in C_j}\hat{P}(x_i)\\&=\sum_{i\in C_j}\left(\log\frac{1}{\sqrt{2\pi}\hat{\sigma}^d}-\frac{1}{2\hat{\sigma}^2}\|x_i-x_j\|^2+\log\frac{n_i}{n}\right)\\&=-\frac{n_j}{2}\log(2\pi)-\frac{n_j\cdot d}{2}\log(\hat{\sigma}^2)-\frac{n_j-k}{2}+n_j\log n_j-n_j\log n\end{aligned} \tag{18}$$

最后,BIC 可以由公式(19)给出:

$$BIC(M_i)=\sum_{j=1}^{k}\hat{l}(C_j)-\frac{p_i}{2}\cdot\log n \tag{19}$$

给定一个候选模型集,其中具有最大 BIC 值即 $\text{argmax}_i BIC(M_i)$的模型被选中.我们利用 BIC 来测量话题聚类过程中话题数量是否优化,亦即当两个类需要合并时,计算局部的 BIC 值,如果合并以后的话题结构的 BIC 值大于合并以前的 BIC 值,则进行话题合并,否则不进行话题合并.局部的 BIC 值只涉及当前需要合并的这部分话题,在有些工作中[14]还进行全局的BIC值计算,即当结构发生变化时,都还计算整个结构的BIC值.本文后面的实验将给出:局部 BIC 值就可以获得全局 BIC 值近似的结果,但是局部 BIC 值比全局 BIC 值的计算复杂度要少很多.

2.4 增量型的话题聚类

在聚类算法的开始,每个新闻文档表示一个类,在聚类的每一步,选择最近的两个类进行合并.事实上,每一个话题都可能包含不同的观点或者不同的叙述重点,因此,每个话题可以由多个子话题构成,每个子话题表示一个观点或者叙述的重点.对于每个话题,在本文中都用 c 个子话题组成,也就是在话题聚类的过程中每个类都由 c 个子类组成.子话题在聚类中的另一种解释是:当把每一个新闻文档当做特征向量空间的一个节点时,这样在类的节点空间中存在一些局部节点密度大的部分,这些处于局部节点密度大的节点集组成一个子类,这样一个类就可以按照密度分布被划分为数个子类.

由于完全正确地选择所有局部密度大的子集是一个非常复杂的问题,因此我们利用一个启发式的策略选择子类,该过程是:首先在类中选择 c 个分散的节点,也就是选择离中心节点最远的且相互之间距离最大的 c 个节点,然后以这些节点作为中心节点,选择离他们最近的节点作为各自子类的成员,该算法的伪代码如图 1 所示.

每个类的组成由 c 个子类表示后,则类之间的相似度也由不同类的子类之间的相似度来测量,也就是两个类的相似值即为两个类中子类之间最大的相似值.该启发算法的另一个目标是:减轻类的边缘节点对聚类算法的影响,如图 2 所示情况,如果仅以中心向量表示每个类,并且两个类的距离由两个类的中心向量的距离决定,则类 A 中的 a 节点有可能被误判为类 B 的节点;而采用子类后,a 节点可以属于它所在的由局部节点密集组成的子类.

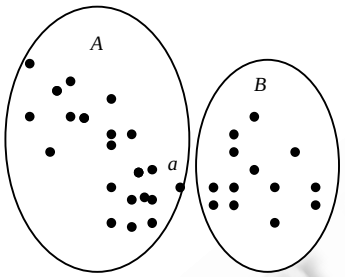

Fig.2 Distribution of nodes in A, B

图 2 A,B 中节点的分布

图 1 给出了话题检测的伪代码,图 3 给出了两个话题类合并过程的伪代码.话题检测算法的输入为新闻文档集 S 和每个类划分子类的个数 c,每个新闻文档表示为特征向量空间的一个点.话题聚类的开始,每个文档被当作一个独立的类,每一步最近的两个类被合并为一个新类,直到没有类可以再被合并.最后,每个类表示一个话题.

```
Procedure merge(X,Y,c)
 {
 1.   Z=X∪Y;
 2.   Z.mean= (|X|X.mean+|Y|Y.mean)/(|X|+|Y|);
 3.   tempSet=∅;
 4.   for (i=1 to c do){
 5.   maxDist=0;
 6.   for each point p in X.subpoints∪Y.subpoints do{
 7.   if i=1;
 8.     Dist=dist(p,Z.maen)
 9.   else
 10.    Dist=min{dist(p,q):q∈tempSet}
 11.   if (Dist≥maxDist){
 12.     maxDist=Dist;
 13.     maxpoint=p;
 14.   }}
 15.   tempSet=tempSet∪{maxpoint}
 16.  }
 17.  Z.subpoints=tempSet;
 18.  sub_clusters=build_subcluster(tempSet);
 19.  for each point p in Z do
 20.    q=p's nearest point in tempSet;
 21.    allocate p to q's sub_cluster;
 22.    computing sub_mean of each sub_cluster;}
 23.  return Z;}
```

Fig.3 Procedure of merging clustering

图 3 类合并过程伪代码

每个类 X 都保存了类中所有的节点,$X.mean$ 和 $X.subpoints$ 分别表示类的中心向量和 c 个分散在类中的节点,这 c 个节点是离类中心最远而且相互之间也尽量远.$dist(X,Y)$表示两个类之间的距离,距离可以是任何形式的距离,在本文,我们用 *cosine* 计算距离,也就是用两个类的中心向量之间的 *cosine* 值表示两个类之间的距离.

因此,两个类之间的距离由如下公式计算:

$$dist(X,Y) \triangleq \min_{x\in X.subtopics, y\in Y.subtopics} dist(x,y) \tag{20}$$

在图 1 中,第 2 行表示建立一个类似堆栈的数据结构 *ST*,*ST* 的元素为类对信息,并且按类对之间的相似度降序排列,当 *ST* 为空时表示话题检测结束.在第 5 行,一对最相似的类从堆栈中的头部元素提取出来,在第 6 行中它们被预合并为一个类.第 7 行,BIC 值用来决定这两个类能否被最终合并,即当连个类合并后的 BIC 值大于合并前的 BIC 值,则进行合并,否则不合并.如果两个类被合并为一个新类,则 *ST* 中的元素被更新,新类 *Z* 与其他类组成 *ST* 元素被插入到 *ST* 中相应的位置,类 *X* 和 *Y* 所在的类对被删除,第 9 行为特征向量的增量型提炼.

图 3 中的第 4 行~第 16 行为类合并操作,它是一个 for 循环操作.首先选择一个离中心最远的节点,在接下来的每次循环操作中,从 *Z* 中选择一个离以前选择的节点最远的节点,然后选中的节点被用来构建 *c* 个子类.开始时,这 *c* 个子类都只包含一个节点,对 *Z* 中的每个节点 *p*,从 *c* 个刚选出的节点中找出离它最近的节点,再把 *p* 分配到该节点所在的类.由于在聚类开始时新闻文档之间的相似度矩阵被计算好并且保存,因此子类的建立过程时间开销不大,时间复杂度为 $O(c)$.

2.5 话题聚类中的预聚类

在本文的话题聚类算法中,首先需要计算文档库中文档之间相似度矩阵.这样,随着文档数量的增加,相似度矩阵计算量会变得非常巨大,而整个话题检测算法的复杂度为 $O(n^2\log n)$,严重影响了算法的实用性.实际情况表明,新闻事件都有一个持续时间的特性,即对一个话题事件的报道基本上都集中在一个时间段内,之后,关于该话题的报道变得很少甚至没有.一个越“旧”的话题越不可能包含刚产生的新闻报道,而新产生的报道属于最近的话题的可能性更大.因此,在话题聚类过程我们可以利用这个特性来减少算法的时间复杂度.如图 4 所示为 TDT-4 语料中话题持续时间的分布图,话题持续时间大部分少于 4 个月.下面,我们引入对顺序输入的文档预聚类的方法来减少文档量大的情况下算法的复杂度.

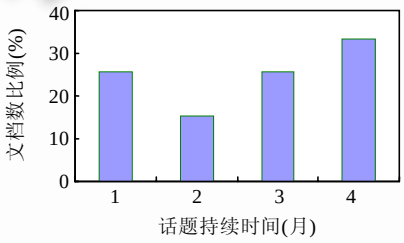

Fig.4 Distribution of topic duration

图 4 话题持续时间分布图

预聚类的基本思想是:新闻文档都有发生的时间,以时间段为单位,把发生在同一时段 Θ 内的文档划分为一个子集,这样,整个文档被划分为 *n* 不相交的子集.在每个子集内进行聚类运算,直到子集中将要合并的两个类之间的相似度大于一个阈值 θ 时停止.最后,对所有的类进行本文提出的聚类运算.只要保证阈值 θ 适当的小,即使一个话题中的文档在预聚类时被划分到不同的子集中,在最后的聚类当中还是可以被聚到一个话题类中.因此,预聚类的方法不会太多地影响话题检测的性能.在本文的实现中,Θ 取 2 个月.从图 4 可以看出,Θ 可以覆盖大部分话题所持续的时间.实验结果表明:预聚类的方法能够大幅度减少话题检测算法的运行时间,而且仍然能够获得较高的话题检测性能.

3 实验及结果分析

在本文的实验中,我们利用 TDT-4 语料库作为实验对象,并且实现了其他两个话题检测方法作为与本文方法的对比.

3.1 性能评价指标

我们选用 TDT-4 中的英文 story 作为实验对象,共包含大约 2 300 个 story 文档,收集的时间为 2000 年 9 月~2001 年 1 月.话题检测的一个主要评价指标就是新闻文档集中文档被分配到正确话题的优劣程度,显然,越多的文档被分配到正确的话题则性能越好,越多的话题被识别出来则性能越好.在本文中,每一个话题都用一个类表示,下面描述中就以类来代替话题.由于聚类输出的类与实际中的类不存在直接的对应关系,因此需要给输出的类标记代表实际类的标签,标记方法是:输出的类与实际中那个类匹配的最好,则标记上该类的标签,而匹配的程度由两个类中所含相同文档的个数决定[1].

信息检索中广泛采用表 1 来评价聚类算法,这个表是个表示类-话题 2-2 的双向表,其中,a,b,c 和 d 分别表示 4 种情况下文档的数量.为了评价需要,*Recall*,*Precision*,*Miss*,*False Alarm*,*F*1 在下面给出定义:

- $Recall=a/(a+c)$, if $(a+c)>0$;否则,无定义;
- $Precision=a/(a+b)$, if $(a+b)>0$;否则,无定义;
- $Miss=c/(a+c)$, if $(a+c)>0$;否则,无定义;
- $False\ Alarm=b/(b+d)$, if $(b+d)>0$;否则,无定义;
- $F1=2*Recall*Precision/(Recall+Precision)$.

Table 1 A cluster-topic contingency table

表 1 类-话题双向矩阵

	In topic	Not in topic
In cluster	*a*	*b*
Not in cluster	*c*	*d*

为了评价本文提出的话题检测方法的性能,我们实现了下面 3 个话题检测系统来进行比较:

系统 1(*K*-means),这个话题检测系统中 *K*-means 聚类算法对新闻文档进行聚类,聚类输出的类表示话题;

系统 2(CMU),该系统实现的话题检测方法主要步骤如下:首先,利用训练语料建立所有词语的倒排文档频率(IDF),当新的新闻文档在规定的迁移时间窗口内到达时,词语的 IDF 被更新.然后,迁移窗口内的文档利用分段的 GAC 方法进行聚类,该聚类方法把文档集按照时间顺序划分成连续的桶(buckets),桶表示发生在一段时间内的 story 文档集,不同的桶之间没有交集,所有的桶的联合等于 TDT-4 corpus.在每个桶中,利用 cosine 相似值对文档进行聚类,最后合并桶中的类;

系统 3(TPIC),该系统实现本文提出的话题检测方法,该话题检测方法采用局部 BIC 值,同时,为了对比局部 BIC 与全局 BIC 的性能,我们在第 3 个实验中实现了基于全局 BIC 的话题检测方法(TPIC2),该方法利用全局 BIC 值来决定类的合并.

3.2 实验及分析

在第 1 个实验中,我们测试比较 3 个系统的 *Recall*,*Precision*,*Miss* 以及 *F*1.在本实验中,我们设定所有系统的类别数目为 TDT-4 中标记的话题的数目,所有的系统都运行多次,每次实验结果都显示 TPIC 具有最优的效率.图 5 给出了 3 个系统的多次实验结果的平均值的比较.从图中数据可以看出,本文提出的话题检测方法性能明显优于其他两个话题检测方法,因为 TPIC 利用了子话题的思想,并且提取了具有高辨别能力的特征词,减少了其他系统中基于类的中心向量方法对话题检测性能的影响,减少了话题检测的误差,提高了话题检测的准确率.

第 2 个实验主要比较各系统在不同文档数量情况下话题检测的时间消耗,文档为语料库中随机选择的.图 6 给出了不同系统话题检测方法运行所需时间的比较,由图可以看出,TPIC 相对其他方法,在相同文档集的情况下需要较少的运行时间.这是由于在 TPIC 中利用话题的时间段特性增加了预聚类的步骤,减少了话题检测所需的时间.

在第 3 个实验中,我们将测试 TPIC 自动检测话题数量的性能,即检测出的话题数量与真实话题数量的差异情况.我们还利用 *X*-means 算法来进行比较,*X*-means 是 *K*-means 算法的扩展,它利用了 BIC 值在聚类的过程中

估计类的个数,具体的实现见文献[13].实验结果的具体情况如图 7 所示,从图中可以看出,TPIC 在话题数量估计方面并不比 *X*-means 差,有些情况下效果还好,与真实话题数量的差异也在 10%以内.另一方面,结果数据表明,全局 BIC 值与局部 BIC 值对于话题数量检测来说效果差不多,但是局部 BIC 值比全局 BIC 值计算要简单得多.因此,采用局部 BIC 值的 TPIC 能够以较少的开销检测话题的数量.

	K-means	CMU	TPIC
Recall (%)	50	62	80
Precision (%)	48	82	84
Miss (%)	50	35	12
False Alarm (%)	0.25	0.13	0.155
*F*1	0.59	0.70	0.84

Fig.5 Topic detection results

图 5 各系统话题检测结果

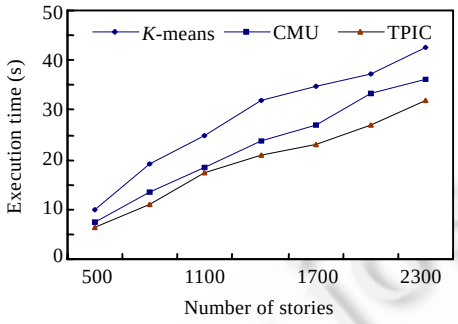

Fig.6 Comparison of execution time

图 6 系统运行时间比较

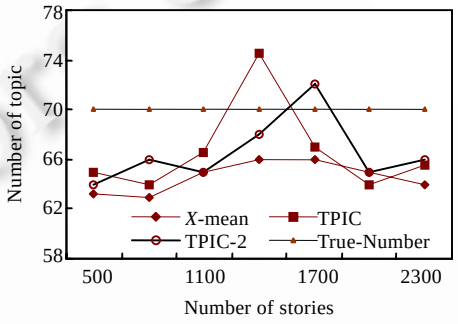

Fig.7 Comparison of topics number between 3 systems

图 7 系统话题数量对比

4 结 论

话题检测任务的一个重要问题就是怎样把文档按照讨论的话题把他们聚类到相应的话题中,当前已有许多基于聚类的话题检测方法,但是它们仅利用了聚类算法,没有利用话题本身的特点,存在很多不足之处,性能有待提高.本文提出了一种基于增量型聚类的自动话题检测方法,该方法能够在话题检测的过程中不断提取具有话题辨别能力的特征,利用改进的权重计算方法给特征赋予权重值,以及通过引入子话题的方法提高话题检测的准确率,利用话题的时间段特性提高话题检测的效率,利用局部 BIC 值估计话题的个数.实验结果表明,该方法与其他话题检测方法相比具有更高的召回率、准确率和 *F*1 值等,能够以更快的速度检测话题,并且能够以较小的误差检测话题的数量.

References:

[1] Seo YW, Sycara K. Text clustering for topic detection. CMU-RI-TR-04-03. Pittsburgh: Robotics Institute, Carnegie Mellon University, 2004. 1−11.

[2] Allan J, Carbonell J, Doddington G, Yamron J, Yang Y. Topic detection and tracking pilot study final report. In: Roukos S, ed. Proc. of the Defense Advanced Research Projects Agency Broadcast News Transcription and Understanding Workshop. Virginia: Academic Press, 1998. 194−218.

[3] Yang Y, Carbonell J, Brown R, Pierce T, Archibald B, Liu X. Learning approaches for detecting and tracking news events. IEEE Intelligent Systems, 1999,14(4):32−43. [doi: 10.1109/5254.784083]

[4] Giridhar K, Allan J. Text classification and named entities for new event detection. In: Järvelin K, ed. Proc. of the 27th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2004. 297−304. [doi: 10.1145/1008992.1009044]

[5] Wang ZM, Zhou XS. A topic detection method based on bicharacteristic vectors. In: Proc. of the Int'l Conf. on Networks Security, Wireless Communications and Trusted Computing. Vol. 2. Washington: IEEE Computer Society, 2009. 683−687. [doi: 10.1109/NSWCTC.2009.153]

[6] Jo YY, Lagoze C, Giles CL. Detecting research topics via the correlation between graphs and texts. In: Caruana R, ed. Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2007. 370−379. [doi: 10.1145/1281192.1281234]

[7] Griffiths TL, Steyvers M. Finding scientific topics. Proc. of the National Academy of Sciences of the United States of America, 2004,101(Suppl 1):5228−5235. [doi: 10.1073/pnas.0307752101]

[8] Wang X, McCallum A. Topics over time: A non-Markov continuous-time model of topical trends. In: Craven M, ed. Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2006. 424−433. [doi: 10.1145/1150402.1150450]

[9] Blei DM, Lafferty JD. Correlated topic models. In: Schölkopf B, ed. Proc. of the Advances in Neural Information Processing Systems. Vancouver: MIT Press, 2006. 123−130. [doi: 10.1145/1143844.1143859]

[10] Erosheva E, Fienberg S, Lafferty J. Mixed-Membership models of scientific publications. Proc. of the National Academy of Sciences, 2004,101(Suppl 1):5220−5227. [doi: 10.1073/pnas.0307760101]

[11] Mei Q, Zhai C. Discovering evolutionary theme patterns from text—An exploration of temporal text mining. In: Bayardo R, ed. Proc. of the 11th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, 2005. 198−207. [doi: 10.1145/1081870.1081895]

[12] Flynn C, Dunnion J. Topic detection in the news domain. In: Waldron J, ed. Proc. of the Int'l Symp. on Information and Communication Technologies. Trinity College Dublin: ACM Press, 2004. 103−108.

[13] Pelleg D, Moore A. *X*-Means: Extending *K*-means with efficient estimation of the number of cluster. In: Langley P. Proc. of the 17th Int'l Conf. on Machine Learning. San Francisco: ACM Press, 2000. 727−734.

[14] Kruengkrai C, Sornlertlamvanich V, Isahara H. Refining a divisive partitioning algorithm for unsupervised clustering. In: Abraham A, ed. Proc. of the 3rd Int'l Conf. on Hybrid Intelligent Systems. Melbourne: IOS Press Amsterdam, 2003. 535−542.

[15] Liu YF, Qi H. A Modified weight function in latent semantic analysis. Journal of Chinese Information Processing, 2005,19(6): 64−69 (in Chinese with English abstract).

附中文参考文献:

[15] 刘云峰,齐欢.潜在语义分析权重计算的改进.中文信息学报,2005,19(6):64−69.

张小明(1980－),男,湖南耒阳人,博士生,主要研究领域为 P2P,话题检测与跟踪.

巢文涵(1979－),男,博士,讲师,主要研究领域为机器翻译,自然语言处理,数据挖掘.

李舟军(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为高可信软件技术,信息安全技术,智能信息处理技术.