第41卷 第4期 西 安 交 通 大 学 学 报 Vol.41 №4
2007年4月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Apr. 2007

一种增量式文本软聚类算法

冯中慧，鲍军鹏，沈钧毅
（西安交通大学电子与信息工程学院，710049，西安）

摘要：针对传统文本聚类算法时间复杂度较高，而与距离无关的算法又不适用于动态、变化的文本集等问题，提出了一种基于语义序列的增量式文本软聚类算法.该算法考虑了长文本的多主题特性，并利用语义序列相似关系计算相似语义序列集合的覆盖度，同时将每次选择的具有最小熵重叠值的候选类作为一个结果聚类，这样在整个聚类的过程中大大减小了文本向量空间的维数，缩短了计算时间.由于所提算法的语义序列只与文本自身相关，所以它适用于增量式聚类.实验结果表明，算法的聚类精度高于同条件下的其他聚类算法，尤其适合于长文本集的软聚类.

关键词：语义序列；增量式聚类；软聚类；文本聚类
中图分类号：TP18 **文献标识码**：A **文章编号**：0253-987X(2007)04-0398-04

Incremental Algorithm of Text Soft Clustering

Feng Zhonghui, Bao Junpeng, Shen Junyi
(School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: Focusing on the problems that the text clustering has high time complexity, the algorithms that are independent on the distance are unsuitable for dynamic and changing corpus, and the multi-subject characteristics of a single text cannot be considered in traditional algorithms, an incremental algorithm of text soft clustering based on semantic sequence is proposed, in which the clustering candidate with minimum entropy overlap value is selected as a result cluster by using similarity relation of semantic sequences and calculating the coverage of similarity semantic sequences set. The dimensions of text vector space are decreased dramatically in the clustering procedure, so the computing time can be reduced. Since the semantic sequence is only related to text, it is available for incremental clustering. The comparison of experimental results shows that the algorithm can achieve higher precision than other algorithms under same conditions, especially for soft clustering of long texts set.

Keywords: semantic sequence; incremental clustering; soft clustering; text clustering

传统的文本聚类算法通常是建立在对向量空间模型进行距离、相似度计算的基础上的，具有相当高的时间复杂度，而性能却随复杂度的增加急剧下降.因此，研究者提出了与距离无关的算法，并设法优化文本的表示方法和文本的聚类性能[1-3]，这些算法均取得了较好的效果，但不适用于动态、变化的文本集聚类.

当文本集处于动态变化的过程中时，就必须通过修改文本集的聚类结果来反映这种变化.因此，急需研究有效的增量式文本聚类算法.另外，硬聚类方法（一个文本只分配一个类别）由于没有包括足够的聚类结果，所以也无法全面地体现文本信息.软聚类方法能按照不同的主题把同一个文本分配到不同类别，从而可从多个角度看待文本，以发现文本的多种特征.

由于语义序列仅仅包含文本自身的信息，不包

收稿日期：2006-07-05. 作者简介：冯中慧（1977～），女，在职博士生，讲师；沈钧毅（联系人），男，教授，博士生导师.
基金项目：国家自然科学基金资助项目(60673087).

含文本集范围上的全局信息,所以它适于在动态、变化的文本集上进行增量式文本聚类.此外,语义序列可体现文本局部信息,从而能够得到文本之间的更多、更详细的共同点,尤其适合长文本集的软聚类.

本文利用了语义序列,提出了一种基于语义序列的增量式文本软聚类算法 (ISSSTC).

1　基本概念

令 $D=\{d_1,\cdots,d_k,\cdots,d_n\}(1\leqslant k\leqslant n)$表示要聚类的文本集合,每个 d_k 是一个文本;s 表示 d_k 中的一个单词序列,即 $s=w_1w_2\cdots w_m$,其中 w_i 表示在 s 中 $i(1\leqslant i\leqslant m)$位置上的单词.文献[4]给出了一种语义序列的定义,其语义序列主要用于文本复制检测,而本文的语义序列则用于文本聚类,因此需对文献[4]的定义进行修改,使其适用于大规模文本集.

定义 1　在 s 中,位置 i 上的单词距离 $\sigma(i)$就是从单词 w_i 到单词 w_h 之间的单词数目,记为 $\sigma(i)=i-h$,其中 $w_i=w_h$ 并且 $w_i\neq w_k(1\leqslant h<k<i\leqslant n)$.也就是说,$w_i$ 和 w_h 是同一个单词而且在 w_i 与 w_h 之间的任何单词都不同于 w_i 或 w_h.如果 w_h 不存在,即 w_i 是第一次出现,那么定义在 s 中 i 处的单词距离为无穷大,即 $\sigma(i)=\infty$.

定义 2　在 s 中,位置 i 上的语义密度 $\rho(i)$就是该处单词距离的倒数,记为 $\rho(i)=1/\sigma(i)$.

事实上,一个文本可以看成是一个很长的单词序列,$\sigma(i)$就是在单词序列 s 中第 i 处的单词距离,如果这个距离很短,那么就说明在某个段落单词 w_i 频繁出现,即在这个段落中单词 w_i 的密度比较大.因此,可以认为在一个局部范围内,高密度的单词反映了这个范围内的局部语义特征.

定义 3　s 的一个子序列称之为语义序列 $s[i]=w_{i_1}w_{i_2}\cdots w_{i_r}$,其中 $i=[i_1,i_2,\cdots,i_r]$ $(1<i_1<i_2<\cdots<i_r\leqslant m)$.该子序列必须满足以下条件:

(1) $|s[i]|>1$

(2) $0<i_{k+1}-i_k\leqslant\varepsilon$,　$1\leqslant k<r$

(3) $\rho(i_k)\geqslant\delta$,　$1\leqslant k\leqslant r$

(4) $(i_1-\varepsilon\leqslant i_h<i_1)\rightarrow\rho(i_h)<\delta$,　$(i_r<i_j\leqslant i_r+\varepsilon)\rightarrow\rho(i_j)<\delta$

(5) $(\forall w_{i_k},w_{i_l}\in s,i_k\neq i_l)\rightarrow w_{i_k}\neq w_{i_l}$,　$1\leqslant k\leqslant r$,　$1\leqslant l\leqslant r$

其中 δ 和 ε 是用户可以调节的参数.

s 中的一个 $s[i]$就是将 s 中的低密度单词删除之后剩下的一个连续的单词序列,这里的连续是指:在 $s[i]$中相邻的 2 个单词,它们在原始文本 s 中的位置之差不应该大于给定的阈值 ε.条件是:①要求一个 $s[i]$是 s 的一个非空的子序列;②$s[i]$中单词之间应具有连续性;③$s[i]$中的每一个单词都是局部频繁的;④要求在一个 $s[i]$之前和之后必须存在一段低密度区间;⑤保证 s 中不存在重复的单词.条件④是限制 $s[i]$使其不能无限扩展,即一个 $s[i]$必须出现在 2 个低密度区域之间,低密度区间又分隔开了不同的 $s[i]$.

2　聚类过程

从语义序列的定义可知,满足条件的高密度单词序列(出现频率高)能够反映局部范围内的语义特征.从这个方面来看,语义序列与关联规则中的频繁项集有些类似,所以本文在借鉴 Florian 等人[5]设计的文本聚类方法的基础上,提出了基于语义序列的文本聚类.

2.1　相关定义

定义 4　度量 2 个语义序列 $s[i]$、$s[k]$的相似程度

$$\mathrm{sim}(s[i],s[k])=\frac{|s[i]\cap s[k]|}{\min(|s[i]|,|s[k]|)}$$

式中:$|\cdot|$是 s 中的单词数目.

定义 5　给定文本 d_k,其所包含的所有语义序列构成语义序列集合 $\Omega(d_k)$;给定文本集合 D,D 中文本的 $\Omega(d_k)$并集构成

$$\Omega(D)=\bigcup_{k=1}^{n}\Omega(d_k)$$

每一个 $s[i]$都反映了文本在某一局部范围内的语义特征,把所有局部特征集合在一起也必定会体现出某种全局特征.因此,可以认为 $\Omega(d_k)$不但包含了一篇文本的局部语义特征,而且还包含了一定的全局特征.

定义 6　将 $\Omega(D)$的一个子集称之为相似语义序列集合

$$S[j]=\{s_j[1],\cdots,s_j[i],\cdots,s_j[p]\}(1\leqslant i\leqslant p)$$

其满足以下条件:

(1) $|S[j]|>1$

(2) $(\forall s_j[i]\in S[j],\exists s_j[k]\in S[j],i\neq k)\rightarrow\mathrm{sim}(s_j[i],s_j[k])>\omega(1\leqslant i\leqslant p,1\leqslant k\leqslant p)$

其中 ω 是用户指定的参数,但条件是:①要求一个相似语义序列集合 $S[j]$是 $\Omega(D)$的一个非空子集;②保证 $S[j]$中的每一个语义序列至少与 $S[j]$中另一个不同的语义序列相似.

定义 7　支持 $S[j]$所有文本的集合,即至少包含 $S[j]$中某个语义序列 $s_j[i]$的文本集合称之为覆盖度

$$\mathrm{cov}(S[j])=\{d_k\in D\mid s_j[i]\subseteq d_k$$

$$\wedge\ s_j[i] \in S[j]\}$$

令

$$S=\{S[1],\cdots,S[j],\cdots,S[o]\}\quad (1\leqslant j\leqslant o)$$

表示语义序列集合所构成的一个集合，其每个 $S[j]$ 是一个相似语义序列集合并且 $\mathrm{cov}(S[j])$ 满足最小支持度 λ，而 $\lambda(0\leqslant\lambda\leqslant 1)$ 是一个给定的阈值，即有

$$S=\{S[j]\subseteq\Omega(D)\,\big|\,|\mathrm{cov}(S[j])|\geqslant\lambda|D|\}$$

在基于语义序列的文本聚类过程中，可将 S 中的每个 $S[j]$ 所对应的 $\mathrm{cov}(S[j])$ 看成是聚类结果中的一个候选类. 显然，这样所取得的候选类有很多种，这些候选类之间可能存在重叠现象，即相同文本出现在不同的聚类中. 为了能够从众多候选类中选择出相对较优的类，可利用信息论中的概念——熵[6]，来定义类间的重叠性.

定义 8 熵重叠

$$\mathrm{EO}(C_q)=\sum_{d_k\in C_q}-p_k\ln(p_k)$$

式中：$p_k=1/f_k$ 表示 d_k 属于一个指定聚类的概率，f_k 表示 d_k 所包含的语义序列的数目.

熵重叠反映了 C_q 中的文本在所有候选类中的分布，而 $C=\{C_1,\cdots,C_q,\cdots,C_t\}$ $(1\leqslant q\leqslant t)$ 是 D 的聚类结果集合，每个 C_q 是一个类.

当 C_q 中的每个 d_k 仅支持一个语义序列时，熵重叠为最小值 0. 事实上，随着 f_k 的增大，熵重叠的值也单调增加. 由于熵重叠衡量的是每一个相似语义序列集合区分文本的能力，因此熵重叠值越小，表示所选聚类结果的纯度越高，区分文本的能力也就越好. 另外，熵重叠基于候选类的文本分布，所以它能够平等对待所有语义序列集，从而可取得高质量的聚类结果.

2.2 算法描述

针对文本集 D，经过插入操作可产生增量文本集 ΔD，经过删除操作可产生减量文本集 ∇D，则发生变化后的文本集为 $(D-\nabla D)\cup\Delta D$. 实际上，文本集 $(D-\nabla D)$ 中的文本已经完成聚类，这样就只需在 $\triangle D$ 上进行增量式聚类，因此本文算法分为以下 2 段.

段 1：对 D 进行聚类，聚类结果保存到 C 中.

段 2：保留 $(D-\triangle D)$ 中文本的聚类结果，对增量文本集 ∇D 进行聚类.

算法描述如图 1 所示.

```
算法：基于语义序列完成增量式文本软聚类.
输入：文本集合D；最小语义密度值δ；最大相邻单词
      距离ε；最小相似度阈值ω；最小支持度阈值λ.
输出：D对应聚类结果集合C.
方法：
1)    找出D中所有语义序列并保存在Ω(D)中;
2)    for(i=0;i<Ω(D).size();i++){
3)      for(k=i+1;k<Ω(D).size();k++)
4)        If (sim(s[i],s[k])<ω) then continue;
5)        else{S[j].add (s[k]);//相似语义序列存入相似
          语义序列集合S[j]
6)      if(|cov(S[j])|>λ|D|) then{
7)      S.add(cov(S[j]));//S[j]满足最小支持度作为一个
        候选类加入集合S
8)      }
9)    }
10)   for(i=0;i<S.size();i++)
11)     计算S[i]的熵重叠值并保存;
12)   for(i=0;i<S.size();i++) {
13)     选出最小熵值的语义序列集与S[i]进行交换;
14)     C.add(cov(S[i]);
15)     for(k=i+1;k<S.size();k++)
16)       then重新计算S[k]的熵重叠值并保存;
17)   }
18)   if (∇D≠∅) then{
19)   从C中删除∇D中的文本;
20)   从Ω(D)中删除Ω(∇D)中的语义序列;
21)   }
22)   if (△D≠∅) then{
23)   找出△D中所有语义序列并存入Ω(△D)中.
24)   for(i=0;i<Ω(△D).size();i++)
25)     for(k=0;k<S.size();k++)
26)       for(m=0;m<S[k].size();m++)
27)         if(sim(Ω(△D)[i],(S[k])[m])<ω) then continue;
28)         else S[k].add(Ω(△D)[i]);//相似语义序列集合S[k]改变
29)   for(k=0;k<S.size();k++)
30)     if(S[k]发生改变) then 用新的cov(S[k])代替C[k];
31)   return C;
```

图 1 基于语义序列的增量式文本软聚类算法

2.3 算法的时间复杂度分析

本文算法采用语义序列概念，通过一种自然的途径极大地减少了文本向量空间的维数. 和获取语义序列的时间相比，算法花在聚类上的时间可以忽略不计，而从语义序列的定义可知，算法对一篇文本只需扫描一遍就能获取其所有的语义序列，因此该算法的时间复杂度为 $O(n)$，其中 n 是所有文本的数目，一般情况下 n 将随着文本集的增大而增大. 本文算法具有一定的伸缩性，适应于大规模文本集.

3 实 验

3.1 文本集

在实验中，选用 2 个文本集来测试聚类结果，第 1 个是 20Newsgroups(NG20)，第 2 个是国际会议 ICMLC2002 论文集. 之所以选用 NG20 文本集，是

因为 Reuters 文本集中的文本太短，难以获取语义序列. NG20 是一种常用的文本数据集[7]，它收集了来自 20 个新闻组的近 20 000 篇新闻，本文选取了其中的 1 862 篇长度大于 3 kB 的文本以构成第 1 个测试集. ICMLC2002 论文集作为长文本测试集，从中选取 482 篇文本构成第 2 个文本测试集.

3.2　实验结果及分析

将 ISSSTC、K-Means、Single-Link、SOM 等常用的文本聚类算法进行实验比较，同时利用了总体评判的宏平均精度 Macro_Pre、宏平均召回率 Macro_Rec、微平均精度 Micro_Pre 和微平均召回率 Micro_Rec 来评价算法，这 4 种度量计算公式分别为

$$\text{Macro\_Pre}=\frac{\sum_{i=1}^{|C|}P(c_i)}{|C|}$$

$$\text{Macro\_Rec}=\frac{\sum_{i=1}^{|C|}R(c_i)}{|C|}$$

$$\text{Micro\_Pre}=\frac{\sum_{i=1}^{|C|}T_i}{\sum_{i=1}^{|C|}T_i+\sum_{i=1}^{|C|}N_i^{\mathrm{I}}}$$

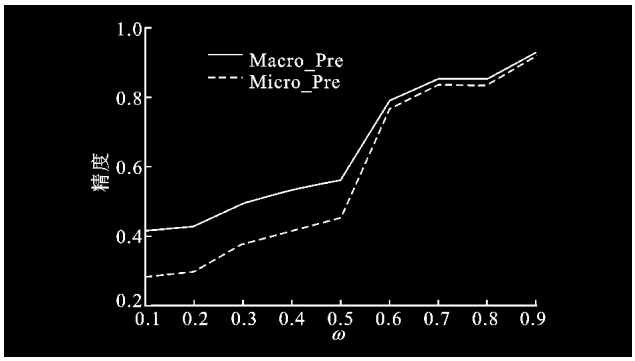
$$\text{Micro\_Rec}=\frac{\sum_{i=1}^{|C|}T_i}{\sum_{i=1}^{|C|}T_i+\sum_{i=1}^{|C|}N_i^{\mathrm{E}}}$$

式中：$P(c_i)$表示类 c_i 的精度；$R(c_i)$表示类 c_i 的召回率；T_i 是指被正确地包含在 c_i 类中的文本；N_i^{I} 是指被错误地包含在 c_i 类中的文本；N_i^{E} 是指被错误地排除在 c_i 类以外的文本.

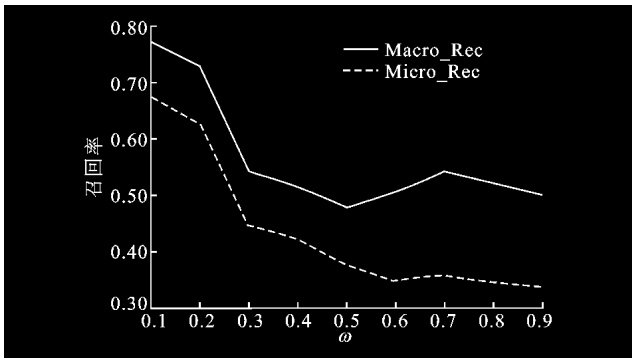
ISSSTC 算法的精度和召回率随相似度阈值 ω 的变化如图 2 所示. 实验中：K-Means、Single-Link 和 SOM 算法的初始参数 $k=20$(NG20)，$k=10$(ICMLC02)，文本向量维数 $\alpha=600$；ISSSTC 算法的初始参数 $\delta=1/30$，$\varepsilon=4$，$\omega=0.8$(NG20)，$\lambda=1/50$(NG20)，$\omega=0.7$(ICMLC02)，$\lambda=1/20$(ICMLC02).

实验结果如图 3、图 4 所示. 从图 3、图 4 可以看出，在这 2 个测试集上，无论是宏平均度量还是微平均度量，ISSSTC 算法的精度均比较高. 究其原因是，ISSSTC 算法的语义序列利用了文本局部信息，所以能更好地体现文本的多主题性，从而获得了文本之间的更多、更详细的共同点和比较高的聚类精度.

此外，从图 2b 看出，ISSSTC 算法的召回率比较低，这因为所提出算法是一种软聚类算法，一个文本可归入多个结果类，而在实验中用来比较结果的标准类是每个文本只归入一个类，这样所获得的实验结果自然有利于硬聚类. 另一方面，自然语言常用同义词、多义词来表达相同或者相似的概念，但内容



(a)精度



(b)召回率

图 2　ISSSTC 算法在 NG20 上的聚类结果

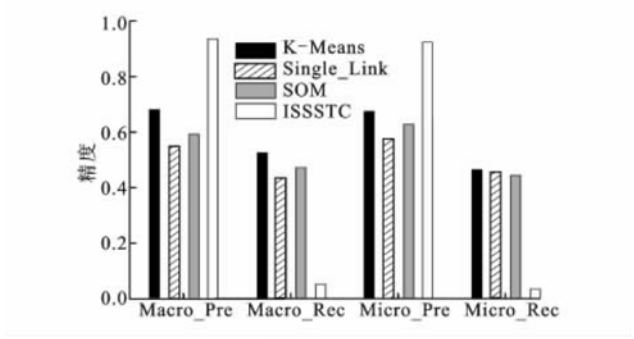


图 3　4 种算法在 NG20 上的聚类结果

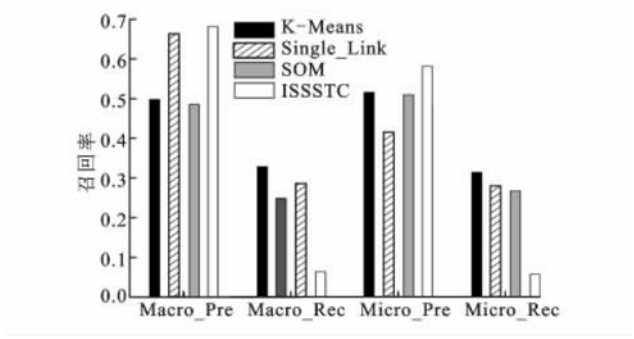


图 4　4 种算法在 ICMLC02 上的聚类结果

相似的文本却不能通过度量语义序列相似度体现出来，从而导致生成了大量规模较小的类，降低了召回率. 在一个向量空间模型中，高维向量可以包含较多的相似内容，在一定程度上抵消了同义词、多义词的影响. 所以，另3种算法的召回率高于ISSSTC算法.

总体来说，使用ISSSTC算法可以提高聚类的精度，尤其适用于长文本集的软聚类.

4 结束语

文本聚类作为一种文本挖掘方法已引起众多研究人员的高度关注. 本文提出了一种有效的基于语义序列的增量式文本软聚类算法——ISSSTC，通过计算相似语义序列集合的覆盖度，每次选择具有最小熵重叠值的类作为一个结果聚类. 实验证明，该算法可提高文本聚类的精度，尤其适用于长文本集的软聚类.

参考文献：

[1] Boley D, Gini M, Gross R, et al. Partitioning-based clustering for web document categorization [J]. Decision Support Systems, 1999, 27(3):329-341.

[2] Zamir O, Etzioni O. Web document clustering: a feasibility demonstration [C]// Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 1998:46-54.

[3] Dhillon I S, Guan Y, Kogan J. Co-clustering documents and words using bipartite spectral graph partitioning [C]// Proceedings of the 7th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2001:269-274.

[4] Bao J P, Shen J Y, Liu X D, et al. Semantic sequence kin: a method of document copy detection [C]// Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin: Springer-Verlag, 2004:529-538.

[5] Beil F, Ester M, Xu X W. Frequent term-based text clustering [C]// Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2002:436-442.

[6] Cover T M, Thomas J A. Elements of information theory [M]. New York: Wiley, 1991.

[7] Slonim N, Tishby N. Document clustering using word clusters via the information bottleneck method [C]// Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2000:208-215.

（编辑 苗凌）