第43卷 第4期 西 安 交 通 大 学 学 报 Vol. 43 №4
2009年4月 JOURNAL OF XI′AN JIAOTONG UNIVERSITY Apr. 2009

利用标签的层次化搜索结果聚类方法

张云，冯博琴
（西安交通大学电子与信息工程学院，710049，西安）

摘要：为了提高搜索引擎返回结果的可浏览性，满足用户对查询质量的要求，提出了一种层次化搜索结果聚类方法. 首先，从搜索引擎的返回结果提取出文档集，并对每一个文档进行词干化、去除停用词等操作. 然后，根据词共现信息来发现文档集中的频繁2元组，再将2元组扩展为n元组，对所有元组进行去冗余、重要性排序，从而获得候选聚类标签. 最后，基于该标签对返回结果中的文档进行分配与聚集，形成层次化聚类结果. 实验结果表明，所提方法可以通过获得的准确、可读性较好的聚类标签，帮助用户有效地浏览搜索引擎返回的结果. 与Vivisimo、STC、Lingo算法比较，以及在多个评价指标上的综合实验结果也表明，该方法是有效的.

关键词：搜索结果聚类；词共现；候选聚类标签；层次化聚类
中图分类号：TP391 **文献标志码**：A **文章编号**：0253-987X(2009)04-0018-04

Clustering Method Based on Label Hierarchical Search Results

ZHANG Yun, FENG Boqin
(School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: A novel clustering method based on hierarchical search results is proposed to facilitate users browsing web search results produced by search engines and to locate the interesting information quickly and efficiently. The snippets are collected and preprocessed. Frequent bigrams are identified based on term co-occurrence information, from which n-grams are obtained. After filtering out the redundant phrases and sorting by significance, candidate cluster labels are obtained. Finally, the snippets are grouped into clusters based on the candidate cluster labels, and a hierarchical result is generated. Experimental results show that the proposed method can generate accurate and highly readable cluster labels, which can help users effectively browse through the search results returned by search engine, and locating their interesting information. The method outperforms Vivisimo, Lingo and STC algorithms on different indexes. A comparison on Chinese dataset further illustrates the validity of the method.

Keywords: search results clustering; term co-occurrence; candidate cluster label; hierarchical clustering

目前，大多数搜索引擎返回搜索结果过于庞大和杂乱，用户难以从中快速找到自己感兴趣的信息. 为了提高搜索引擎返回结果的可浏览性，满足用户对查询质量的要求，一种较好的解决方法是对搜索结果进行聚类，创建主题类别，并对每个类别使用相应的主题词加以描述. 这样，可将不同的主题类呈现给用户，使用户能够在较高的主题层次上浏览搜索引擎返回的结果，从而快速有效定位到感兴趣的信息上.

据此，本文提出了一种新的搜索结果聚类算法，从搜索结果集中抽取有序的、由连续的或不连续的词构成的具有丰富语义信息的短语作为聚类标签，

收稿日期：2008-08-02. 作者简介：张云(1982—)，女，博士生；冯博琴(联系人)，男，教授，博士生导师. 基金项目：国家高技术研究发展计划资助项目(2006AA01Z210).

再基于标签对搜索结果进行分配和聚集，形成层次化的聚类结果.

1　基于标签的层次化搜索结果聚类

本文方法主要包括 3 部分：①搜索结果的获取和预处理，即通过对搜索引擎的返回结果进行网页分析和处理，先提取出 snippet 集（文档集），该集由排在返回结果最前面的 N 个文档构成，再对文档集中的每一个文档进行一般的词干化、去除停用词等操作；②抽取候选聚类标签（本文核心）；③基于抽取得到的候选标签对返回结果中的文档进行分配和聚集，形成层次化聚类结果.

1.1　抽取候选聚类标签

为了从文档集中抽取出具有丰富语义信息的候选聚类标签，本文提出了一种基于词共现的标签抽取方法. 首先根据词共现信息来发现文档集中的频繁 2 元组，然后采用一种基于图的方法将 2 元组扩展为 n 元组. 所谓词共现是指，2 个词在同一句子中且在指定的距离内同时出现. 标签抽取方法分下述 3 个步骤.

步骤 1　抽取有效 2 元组. 如果 2 个相关词是一个有效的 2 元组，那么它们至少需要满足 2 个假设[1]：①2 个相关词频繁出现；②2 个词受句法限制致使在出现时具有一个固定的模式. 基于该假设，通过分析词的分布，本文提出下述的过滤条件，从而将有效的 2 元组提取出来.

(1)分析和抽取词的共现信息. 对每个词 w，记录与之共现的所有词 w_i 及其共现的次数，并要求 w_i 必须与 w 属于同一个句子且句子之间的距离不超过阈值 $d(d>0)$，这样可以得到一个共现的词对，记作$\langle w,w_i\rangle$. 由此，可以得到如图 1 所示的结果. 凡是在同一个句子中与 w 在距离 d 中共现的词 w_i 都将记录下来，包括每个相对位置上的出现次数 p_i^j $(-d\leqslant j\leqslant d, j\neq 0)$、共现的总次数 f_i 以及 w 与 w_i 共现的所有句子 ID（存储在 senIds 中）等.

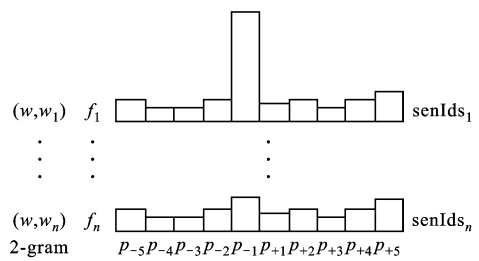

图 1　针对词 w 搜集的词共现信息

(2)过滤词对，提取有效 2 元组. 为了判断$\langle w,w_i\rangle$是否为一个频繁 2 元组，定义了词对强度

$$s=\frac{f-\overline{f}}{\sigma}$$

式中

$$\overline{f}=\frac{1}{n}\sum_{1\leqslant i\leqslant n}f_{ij}\quad \sigma=\Big(\sum_{1\leqslant i\leqslant n}\frac{(f_i-\overline{f})}{n}\Big)^{1/2}$$

s 显现了一个词对的共现频度，适于不同词对之间的比较，可用来过滤共现频度不高的词对. 定义阈值 β_0，低于该强度阈值的词对将被过滤掉.

基于 s 选择频繁词对，还需要分析出具有固定模式搭配的词对作为有效 2 元组. 通过每一个词对中 2 个词位置的出现次数 $p_i^j(-d\leqslant j\leqslant d, j\neq 0)$均值$\overline{p_i}$及其方差$\rho_i$，可描述 p_i^j 在 $2d$ 个相对位置上出现的频率. 将方差定义为词对的广度 E，如果 ρ_i 值比较小，也就是说词 w_i 以均等的概率在 w 周围的任意位置上出现. 相反，如果 ρ_i 值比较大，则意味着 w_i 频繁地出现在 w 周围的某个固定位置上，而这时可认为$\langle w,w_i\rangle$更倾向于一个有效 2 元组. 广度计算式为

$$E=\Big(\sum_{-d\leqslant j\leqslant d,d\neq 0}(p_i^j-\overline{p_i})^2\Big)/2d$$

式中

$$\overline{p_i}=\Big(\sum_{-d\leqslant j\leqslant d,j\neq 0}p_i^j\Big)/2d$$

定义阈值 ρ_0，低于该广度阈值的词对也将被过滤掉，最后剩下的词对可被认为是构成有效 2 元组的词对.

步骤 2　基于 2 元组构造 n 元组. 本文基于图的方法可从 2 元组中发现 n 元组. 利用得到的所有 2 元组构建一个有向图，其中顶点为单个词，边代表连接在一起的 2 个词是一个有效的 2 元组，箭头表示 2 元组的前后位置信息. 因此，存在 w_i 到 w_j 的边 $e(w_i,w_j)$，当且仅当$\langle w_i,w_j\rangle$是一个有效的 2 元组. n 元组$(n>2)$可以被形式化定义为

$$\psi(w_1w_2\cdots w_n)=\begin{cases}e(w_1,w_2) & n=2\\ \psi(w_1w_2\cdots w_{n-1})\wedge\bigwedge\limits_{i=1}^{n-1}e(w_i,w_n) & n>2\end{cases}$$

式中

$$e(w_1,w_2)=\begin{cases}\text{true} & \langle w_1,w_2\rangle\text{ 是有效 2 元组}\\ \text{false} & \text{其他}\end{cases}$$

步骤 3　排序、去冗，得到候选聚类标签. 所得到的所有的 n 元组和 2 元组，以及文档集中出现的单个词（1 元组）都有可能是候选标签. 根据统计特点为每一个元组定义重要性 S，再依 S 排序. 本文改进了经典加权方法，并用它来定义一个元组的重要性. 因

为当一个元组中包含的词数较多时，词频及文档频信息显然会大大降低. 另一方面，一个较长的短语具有较好的描述和表达能力. 因此，定义一个元组的重要性函数

$$S(p) = t(p)B(p)$$

式中

$$t(p) = T_F(p)\lg\left(1+\frac{N}{D_F(p)}\right)$$

$$B(p) = \begin{cases} c^{|p|} - b & |p| \leqslant 8 \\ 5 & |p| > 8 \end{cases}$$

其中：$t(p)$是经典的基于词频-文档频的短语加权方法；$T_F(p)$、$D_F(p)$分别为元组 p 的词频与文档频信息；$B(p)$是基于元组 p 中包含的词数的增强函数；c 与 b 都是常量；$|p|$表示元组 p 中包含的词数. 显然，当一个短语中包含的词数越多，语义信息就越丰富，描述能力越强. 我们采用一个指数函数来增强一个长元组的重要性，同时考虑到了一个元组包含大量词汇的情况，因此定义一个语义增强能力上界，即当元组中包含词的个数超过 8 时，语义增强能力保持在 5. 实验系统中，设定 $c=1.25$，$b=0.5$.

应注意到，当一个词被抽取出来时，它可能存在于包含该词的 2 元组甚至 n 元组中. 假设短语(元组)p_1 与 p_2 具有包含关系，p_2 中所有的词都出现在 p_1 中，则将 p_1 与 p_2 的权重比值作为过滤条件，当比值超过阈值 ω_0 时，删除冗余短语 p_2，即如果 $S(p_1)/S(p_2)>\omega_0$，删除元组 p_2.

去除了冗余元组之后，将剩余的元组根据重要性排序，选择最前面的 M 个元组作为有效的候选聚类标签进行后续的操作.

1.2 搜索结果聚类与层次化

得到候选聚类标签后，基于这些标签对文档集进行聚类和层次化操作，步骤如下.

步骤 1 根据候选标签创建基类. 通过抽取得到的候选标签构建基类，即共享同一个短语的文档构成一个基类，共享短语作为该类的标签. 这样，由 M 个候选标签得到 M 个基类.

步骤 2 基类合并与层次化. 层次化的聚类结构能够帮助用户浏览返回结果. 通过对任意 2 个基类中共享文档的个数来判断基类是否需要进行合并或者满足父子关系. 本文采用类似 SHOC 方法中的基类合并算法，详细内容可参考文献[2].

2 实验分析

2.1 实验设置

分别选取 Vivisimo、Lingo[3] 及 STC[4] 算法与本文方法进行实验比较，实验基于 Carrot2(http://project.carrot2.org/)开源框架来实现. Vivisimo (http://vivisimo.com/)是一个商业的搜索结果聚类引擎，是目前效果比较好的一种算法.

在搜索结果聚类评价中，尚没有一个标准的测试集. 本文基于 Vivisimo 与 ODP (open directory project)(http://www.dmoz.org)构建了评价数据集.

根据文献[5]的分类，我们选择了 3 类 14 个查询词，如表 1 所示.

表 1 查询词

查询类型	查询词
歧义查询	jaguar, apple, matrix, java
实体名称查询	Iraq, London, harry potter, world cup, Clinton, Mozart
一般查询	ipod, travel, health, flower

依次将查询词提交给 Vivisimo，并限定检索数据源为 ODP，最后将返回结果下载下来. 14 个数据集共 2 128 篇文档，平均每个查询词返回 152 篇文档. 对返回结果进行处理，再将构造每一个返回结果集在 ODP 中的目录层次作为标准分类. 将返回结果文档分别输入到不同的聚类算法中，可得到聚类结果评价.

用不同算法的聚类结果与标准分类结果之比 r 来评价不同聚类算法的优劣，其中用到了最常用的聚类评价标准 F-度量[6]，使用了归一化补熵(NCE)和归一化互信息(NMI)[7]指标. 为了公平起见，在评价不同算法的聚类结果时，抽取最前面的 K 个聚类结果进行评价，取 $K=5,10,20$. 应注意：Lingo、STC 算法得到的聚类结果是扁平式的，而 Vivisimo、本文算法是层次化聚类，所以在比较时，将层次化算法形成的最高层聚类结果与 Lingo、STC 算法的扁平结构进行比较. 同样地，ODP 中的目录结构是树形的，应选择第一层目录作为标准分类. 最后，对不同算法的执行时间进行对比分析.

通过多次实验来确定各个阈值，阈值设置如下：$N=200$，$d=5$，$\beta_0=1$，$\rho_0=1$，$\omega_0=1$，$M=120$，$t_1=0.6$，$t_2=0.8$.

2.2 实验结果

图 2 显示了 Vivisimo、Lingo、STC 和本文算法的平均 NMI 结果. 从中看到，本文算法在 NMI($K=5,10,20$)上的结果与 Vivisimo 算法相当，二者均

优于 Lingo、STC 算法. 图 3 显示了 4 种算法的平均 NCE(K=5,10,20)结果，其中本文算法效果最好，略优于 Vivisimo、Lingo 算法，远远优于 STC 算法. 图 4 显示了平均的 F-度量值. 从图 4 可以看出：STC 的 F-度量较高，尤其是 K=5,10 的情况；当 K=20 时，Vivisimo 的 F-度量值超过了 STC. 本文算法在 K=5,10 时与 Vivisimo 算法相当，在 K=20 时略略高于 STC 算法.

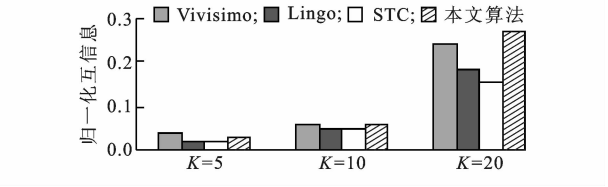

图 2　归一化互信息比较

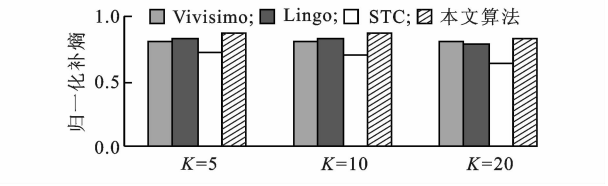

图 3　归一化补熵比较

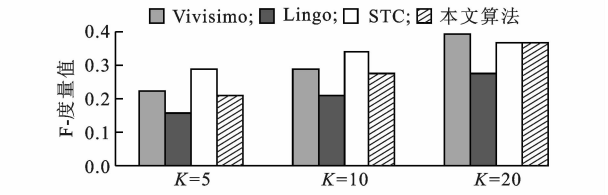

图 4　F-度量值比较

另外，本文算法不仅适用于英文，也适用于中文. 查询词“地震”经百度搜索引擎搜索，再将排列在最前面的 202 个返回结果作为算法输入. 由于 Vivisimo、STC 算法都不能处理中文，在此仅对比分析了 Lingo 和本文算法的聚类情况，结果如图 5 所示.

从图 5 可以看到，Lingo 算法在生成的聚类标签中有很多是没有主题意义的词，例如“版权所有”、“首页”等，虽然有的标签较长，例如“眉山受地震波及影响…”，但是该标签并没有抓获住数据集中的主题，而且每个聚类所包含的网页数(由每个聚类标签后面的数字所示)很少. Lingo 算法虽然可以应用到中文数据集上，但是可以看到应用效果并不好. 对比分析，本文算法的聚类标签由若干个有序的词构成，构成的词组都具有丰富主题语义信息，这表明本文算法的聚类标签是有效的. 图 5 展开了第一个聚类层次，进一步显示了本文算法聚类结果的有效性.

图 5　聚类标签

根据表 1 中给出的查询词重新构造测试数据集，并考察 Vivisimo 的前 100～500 个返回结果. 在一台 8 GB 内存、CPU 4×3 GHz 的 Linux 服务器上，测试了 STC、Lingo、本文算法的平均执行时间，如图 6 所示. 显然，STC 算法的速度是最快的，这与后缀树的快速构建有密切关系. Lingo 算法的执行时间相对来说要长得多，因为它进行了复杂的矩阵计算. 本文算法在网页片段数不超过 200 时速度较快，片段数一旦超过 200，执行时间明显上升，但总体来说，本文算法的执行时间低于 Lingo 算法.

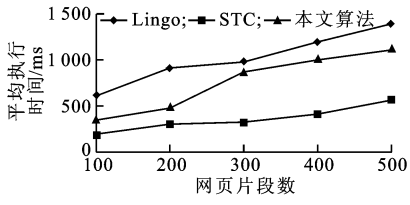

图 6　算法执行时间对比

3 结 论

本文提出了一种新的搜索结果聚类算法，主要贡献有以下几点：①引入了一种新的聚类标签抽取算法，即基于词共现与统计信息抽取频繁2元组，然后用本文的基于图的算法来合并2元组为n元组，最终将得到的有序、由不连续词构成的短语作为聚类标签；②建立了快速的层次化聚类算法，即基于候选标签构造基类，然后对基类进行合并和层次化，由此大大提高了聚类算法的效率，满足在线聚类的要求；③推出了完整的实验评价系统，即基于*Vivisimo*与*ODP*构造基准数据集，并与*Vivisimo*、*STC*、*Lingo*算法进行了比较. 在多个不同指标上的实验结果均表明，本文算法是有效的. 同时，给出了中文数据集上的聚类标签实例，进一步说明算法在抽取聚类标签上的有效性.

由于算法涉及到的阈值较多，如何自动调整阈值，使系统性能最佳是今后的研究内容之一. 另外，我们认为在抽取得到候选聚类标签之后，所提算法的层次化聚类步骤还可以进一步优化.

参考文献：

[1] SMADJA F. Retrieving collocations from text: Xtract [J]. Computational Linguistics, 1993, 19(1): 143-177.

[2] ZHANG Dell, DONG Yisheng. Semantic hierarchical, online clustering of Web search results [C]// Proceeding of the 6th Asia Pacific Web Conference (APWEB). Berlin, Germany: Springer-Verlag, 2004: 69-78.

[3] ZAMIR O, ETZIONI O. Grouper: a dynamic clustering interface to Web search results [C]// Proceedings of the 8th International World Wide Web Conference. Toronto, Canada: Elsevier, 1999: 283-296.

[4] OSINSKI S. An algorithm for clustering of Web search results [D]. Poznan, Poland: Poznan University of Technology, 2003.

[5] ZENG Huajun, HE Qicai, CHEN Zheng, et al. Learning to cluster Web search results [C]// Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2004: 210-217.

[6] 张云，冯博琴，麻首强，等. 蚁群-遗传融合的文本聚类算法 [J]. 西安交通大学学报，2007，41(10)：1146-1150.
ZHANG Yun, FENG Boqin, MA Shouqiang, et al. Text clustering based on fusion of ant colony algorithm and genetic algorithm [J]. Journal of Xi′an Jiaotong University, 2007, 41(10): 1146-1150.

[7] GERACI F, PELLEGRINI M, MAGGINI M, et al. Cluster generation and cluster labeling for Web snippets: a fast and accurate hierarchical solution [J]. Internet Mathematics, 2007, 3(4): 413-444.

（编辑 苗凌）