一种基于关键特征的搜索引擎结果聚类算法

张 辉 谢 科 庞 斌 吴 辉

(北京航空航天大学 开发环境国家重点实验室, 北京 100083)

摘 要: 为了解决用户在搜索引擎结果列表中寻找所需信息困难的问题,帮助用户快速有效地定位有价值的 Web 文档,与向量空间模型方法不同,采用基于关键特征的聚类算法(KFC). 首先从搜索引擎返回结果的关键词里选择重要的词作为关键特征,然后通过分析特征间的关系对特征聚类,最后基于特征聚类结果实现文档的聚类. 通过对实验结果的测试表明了算法的有效性.

关 键 词: 搜索引擎; 算法; 特征提取; 文档聚类; 向量空间模型 VSM; KFC 算法

中图分类号: TP 302.1

文献标识码: A **文 章 编 号**: 1001-5965(2007)06-0739-04

Key-feature-based clustering algorithm for search engine results

Zhang Hui Xie Ke Pang Bin Wu Hui

(National Laboratory of Software Development Environment, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: To solve the problem that users of web search engines are often forced to sift through the long ordered list of document, a new key-feature clustering (KFC) algorithm was presented to help locate the valuable search results that the users really needed, which was different from VSM. The algorithm firstly extracted some key features from the keywords in the search results. Then the relationships between key features were analyzed and features were clustered. Finally, the documents were clustered based on these clusters of key features. The algorithm was tested and validated by the results of experiments.

Key words: search engines; algorithm; feature extraction; document clustering; vector space model; KFC algorithm

目前,搜索引擎结果聚类的研究主要有 3 种方法. 第 1 种方法是利用 Hits, PageRank 算法,分析网页的出、入链接情况,对具有相似特征的网页进行聚类[1]. 此外,曾华军等人在文献[2]中把文本聚类[3]的思想引入到搜索引擎结果聚类中,将搜索引擎的每个结果按照关键词进行切分,得到向量集合,再利用文本聚类的方法进行结果聚类. 第 3 种方法是 Wang Po-Hsiang 等人在文献[4]中提出的,利用用户行为对搜索引擎结果进行聚类,根据用户对结果的选择情况,提取共性,对结果进行优化,将用户感兴趣的结果提前.

目前文本聚类算法大都基于向量空间模型(VSM, Vector Space Model)[5],这类算法易于实现,但是算法并不能明确每个类所代表的主题,仅仅是将文档向量化以后,根据向量之间的相似度大小确定是否将它们归为一类. 这类算法的聚类结果不太准确,且算法的速度比较慢,在实际搜索引擎中很少得到有效应用.

本文在借鉴文本聚类的基础上,以文档的主题发现为算法核心,对结果进行聚类,使用户可以在聚成若干类的结果中方便选择所需信息. 本文提出的结果聚类算法主要思想是:在保证一定聚类效果的前提下,保持较低的算法复杂度.

收稿日期: 2006-08-23

基金项目: 国家科技基础条件平台建设资助项目(2005DKA63901)

作者简介: 张 辉(1968—),男,浙江磐安人,副教授, hzhang@nlsde.buaa.edu.cn.

1 算法思路

本文选取搜索引擎结果中排在前面的若干结果项作为待聚类对象集.对这些结果项进行分词后,得到对应关键词集合,并从这些关键词中提取出作为聚类特征的词(以下简称特征).一般而言,这些特征往往是结果项中出现频率比较高、词性为名词或动词并且与主题紧密相关的一些关键词.若干特征就可代表一个结果项,利用这些特征进行聚类可大大降低计算复杂度.

2 特征选择

一个结果项对应一个关键词向量,该向量记录了每个关键词出现在该结果项中的次数,格式如表1所示.

表1 结果项关键词向量表示格式

关键词	k_1	k_2	…	k_N
出现次数	t_1	t_2	…	t_N

注:$k_i(1\leqslant i\leqslant N)$表示切词后得到的某一个关键词;$N$为该结果项切分得到的不同关键词总数;$t_i(t_i>0)$为关键词$k_i$在该结果项中出现次数.

为了帮助确定特征选取个数,本文对关键词出现在结果项中的分布情况进行了统计,统计数据如图1所示.其中横坐标表示高频特征数量占特征总数量的比重(高频特征个数/总特征个数),纵坐标表示相应的特征出现的词频均值.

从图1可以看出,对特征进行词频从高到低排序后,分布在0.3段以后的特征出现频率接近1,即这些词语大部分只在一个结果项中出现,这样的词语对于主题提取不能提供更多的信息.

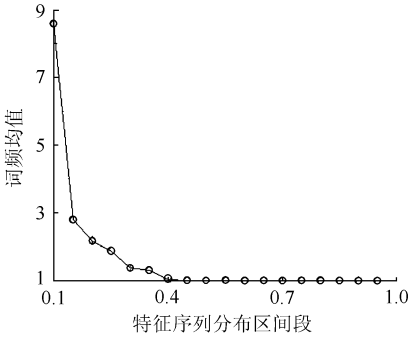

图1 特征序列分布与词频均值关系图

结合图1的统计信息,本文选择排在关键词序列前面数目比例为0.25的关键词为作为一个结果项的特征,它们对于主题提取具有重要意义.

3 自主聚类

3.1 特征间权值计算

本文采用的特征间权值算法借鉴了TF-IDF公式[5],将特征本身的特性和特征间互信息综合考虑进行计算.特征本身的特性用特征出现的频率以及出现该特征的结果项数目来表示;特征之间的关系用不同特征共同出现的结果项数以及在这些结果项中特征各自出现的次数表示.本文主要分析特征在结果项中出现的总次数和包含该特征的结果项数,采用权重取平均值的办法(见式(1)).为了便于计算,现假设如下:

R_{set}为结果项向量集合,$R_{\text{set}}=\{\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_{T_{\text{sum}}}\}$($T_{\text{sum}}$为待聚类的结果项总数);$\boldsymbol{R}_i(1\leqslant i\leqslant T_{\text{sum}})$表示一个结果项向量(如表1所示),$\boldsymbol{R}_i=[(k_1,t_1),(k_2,t_2),\cdots,(k_n,t_n)](n\in\mathbf{N})$;$t_j$ $(1\leqslant j\leqslant n)$表示关键词$k_j$在结果项$\boldsymbol{R}_j$中出现的次数;$S_{\text{feature}}$为提取的特征集合;$f_i,f_j\in S_{\text{feature}}(f_i\neq f_j)$;$T_i,T_j$分别为$f_i,f_j$出现在$R_{\text{set}}$中总的次数;$N_i$,$N_j$分别为在$R_{\text{set}}$中有$k_i,k_j$出现的总的向量数;$N_{ij}$为在$R_{\text{set}}$中$f_i$和$f_j$共同出现的向量数;$T'_i$,$T'_j$分别为$f_i,f_j$在$f_i$和$f_j$共同出现向量中的出现次数;$W_{ij}$为$f_i$和$f_j$间的权值.

根据以上假设,本文的权值计算公式可表述如下:

$$W_{ij}=(F_{f_i}(f_j)+F_{f_j}(f_i))/2 \tag{1}$$

其中

$$F_{f_i}(f_j)=\alpha\times(T'_i/T_i)+\beta\times(N_{ij}/N_i) \tag{2}$$

$$F_{f_j}(f_i)=\alpha\times(T'_j/T_j)+\beta\times(N_{ij}/N_j) \tag{3}$$

$$0\leqslant\alpha\quad\beta\leqslant1\quad\alpha+\beta=1$$

3.2 特征聚类

3.2.1 算法思想

由式(1)计算出所提取的特征集中任意2个特征间的权重,得到集合中元素间的权重集合.在本文中,元素间的关系密切程度是通过权重来表示的,2个元素间的权重越高,则表示这2个元素间的关系越密切;类内元素的密切程度是通过元素的标准差来体现的,标准差越小,则表示该类中元素关系越密切.结合该思想,本文提出了"关键特征聚类"KFC(Key-Features Clustering)算法,该算法将元素间的权值和类内元素间的标准差作为聚类的核心,力求将元素间的权值较大且标准差较小的元素集合构成一个类.

为了描述算法,进行如下符号约定和假设:

$D=\sqrt{\sum_{j=i+1}^{n}\sum_{i=1}^{n-1}\frac{(V_{ij}-\overline{V})^2}{\frac{n(n-1)}{2}-1}}$ 为一个类别的权值标准差,其中,V_{ij} 表示该特征集合$\{f_{S_1},f_{S_2},\cdots,f_{S_n}\}$中的特征 f_{S_i} 和 f_{S_j} 之间的权值,$\overline{V}=\frac{2}{n(n-1)}\sum_{j=i+1}^{n}\sum_{i=1}^{n-1}V_{ij}$,表示该类别已有特征间权值的平均值;$f_{\text{con}}$表示新特征进入类时的权值阈值;$S=\{S_1,S_2,\cdots,S_l\}$表示类别集合,$S_h=\{s_{h1},s_{h2},\cdots,s_{hn}\}(1\leqslant h\leqslant l)$表示一个类别,$s_{hi}(1\leqslant i\leqslant n)$表示类型 S_h 的特征项;S_{temp}为一个类别;D_{temp}表示一个标准差.

算法具体描述如下:

1) 参数初始化:S 为空;

2) 聚类初始化:选取非聚类特征对中,权值最大的 2 个元素对作为初始聚类中心初始化 S_{temp},计算 f_{con}和 D 的值;

3) 挑选新特征:在未被聚类并且未被挑选过的特征集中,选取和类别 S_{temp}中任意一个元素的权值大于 f_{con}的元素,如果存在这样的元素,计算出当其被归为此类后形成新类的标准差 D_{temp},如果 $D_{\text{temp}}<D$,则该元素被归为 S_{temp}类中,并且置 $D=D_{\text{tem}}$,重新设置 f_{con},转向 3),同时,将该元素置为被挑选过;

4) 将类别 S_{temp}添加到类别集合 S 中,如果所有特征均归属到相应的类别中,则聚类结束,并输出归类集 S;否则转向 2).

3.2.2　参数分析和确定

在本算法中,权值阈值 f_{con}的确定成为影响聚类效果的一个关键因素.一般而言,用户输入的关键词越多,表明用户输入的主题信息越明确,相反,关键词越少,则表示结果集合中的主题越不明确,造成主题数过多.因此,用户输入关键词的数目成为一个重要的决定因素.本文在对特征进行聚类的过程中,新特征进入某一类时的权值阈值计算采用如下公式:

$$f_{\text{con}}=g(N_{\text{words_of_query}})*\overline{V}$$

其中函数 $g(N_{\text{words_of_query}})$中 $N_{\text{words_of_query}}$表示用户输入关键词的个数.根据搜索结果,本文对数据进行了如下分析:对所有结果数在 50～100 之间的查询请求进行分类统计,其调节系数值的变化范围是 0.50～0.95,即起始值为 0.50,递增幅度为 0.05,中止范围为 0.95.根据以上实验方案,得出了如下一些统计数据(如图 2 所示.图中横坐标表示一次查询生成的结果聚类类别总数,纵坐标表示出现某类别数时的查询次数).为了避免用户在过长的聚类列表中查找自己感兴趣的类别,本文对生成的聚类类别数有所控制,期望的聚类类别数介于 10～20 之间.

对于 1 个和 2 个查询关键词的情况,统计情况如图 2 所示.从图 2 可以看出,调节系数为 $A=0.60$时,结果聚类具有较好的分布.$A=0.60$ 对应分布曲线的聚类峰值出现在 14～16 之间,和其他峰值出现在这个区域的曲线不同的是,其峰值最高,且曲线在峰值右边变化剧减,即坡度较大,符合系统对聚类的分布期望.

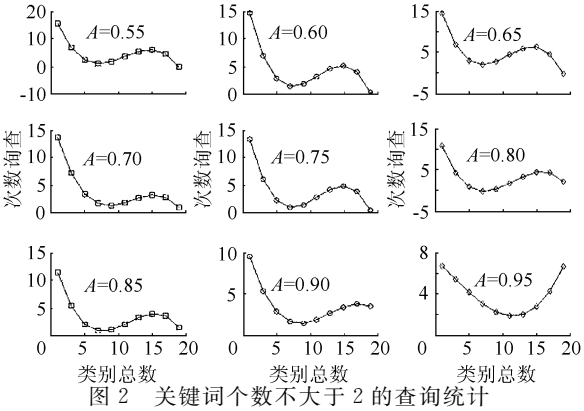

图 2　关键词个数不大于 2 的查询统计

本文还对查询关键词个数为 3 和 4 时进行了类似的分析,当查询关键词个数为 3,$A=0.65$ 时,结果聚类具有较好的分布.当查询关键词个数为 4,$A=0.70$ 时,结果聚类具有较好的分布.

根据以上分析得出如下计算公式:

$$g(N_{\text{words_of_query}})=\begin{cases}0.60 & N_{\text{word_of_query}}\leqslant 2\\ 0.65 & N_{\text{words_of_query}}=3\\ 0.70 & N_{\text{words_of_query}}\geqslant 4\end{cases}\tag{4}$$

4　权值计算

由 3.2 节得出的聚类结果是一个类别集合,其中,每个类别就是一个由关键词组成的集合,本节将计算出由 3.2 节得到的每个特征类别和每个结果项间的权值,这个权值将成为结果项聚类的主要依据.在计算这个权值时,本文根据 TF * IDF[5] 计算公式,并对其进行了一些改进.本算法符号约定如下:$V_{\boldsymbol{R}_i}=V(I_{k_1},I_{k_2},\cdots,I_{k_n})$为结果项 $\boldsymbol{R}_i$ 对应的特征权值向量,其中,I_{ji} 表示结果项 $\boldsymbol{R}_i$ 对应特征 f_j 的权值;$T_{\boldsymbol{R}_ji}$表示特征 f_j 在结果项 $\boldsymbol{R}_i$ 中出现的次数;$T_{\boldsymbol{R}S_i}$为结果项 $\boldsymbol{R}_i$ 中各关键词出现次数总和;N_j 为有 f_j 出现的不同的结果项数;T_{sum}表示结果的总数目;其他符号约定同前.

本文的权重计算公式如下:

$$I_{ji}=\log(1+T_{\boldsymbol{R}_ji}/T_{\boldsymbol{RS}_i})\times\log(T_{\mathrm{sum}}/N_j) \quad (5)$$

5 结果聚类

本文结果聚类的核心是发现主题，帮助用户确定进一步的搜索方向，起到启发式搜索的目的.因此，在本文中，允许一个结果项同属于多个主题.根据第4节中的计算结果，每个结果项同每个特征间有一个权值，每一个权值反映了该特征同目前结果项的关系紧密程度.本文借鉴K-邻近法[6]的思想，选用投票法作为结果聚类的核心思想.本算法中的符号约定和假设如下：$W_{\boldsymbol{RSI}_ihj}\in\{0,1\}$表示结果项$\boldsymbol{R}_i$对应于类别$S_h$中特征$I_{e_{hj}}$的归属权重；$W_{\boldsymbol{RS}_ih}(W_{\boldsymbol{RSI}_ih1},W_{\boldsymbol{RSI}_ih2},\cdots,W_{\boldsymbol{RSI}_ihn})$表示结果项$\boldsymbol{R}_i$对应于类别$S_h$中所有特征的权值序列；$V_{\boldsymbol{RS}_ih}=V(S_h,S_i)\in\{0,1\}$表示结果项$\boldsymbol{R}_i$对应于类别$S_h$的隶属值，值为0表示$\boldsymbol{R}_i$不属于$S_h$，否则，表示属于；$V_{\boldsymbol{RS}_i}=\{V_{\boldsymbol{RS}_i1},V_{\boldsymbol{RS}_i2},\cdots,V_{\boldsymbol{RS}_ih}\}$表示结果项$\boldsymbol{R}_i$对应于类别的权值序列；其他符号约定同前.根据以上假设和约定，具体算法如下：

$$W_{\boldsymbol{RSI}_ihj}=\begin{cases}1 & V_{\boldsymbol{R}_i}\text{中},I_{e_{hj}}\neq 0\\ 0 & V_{\boldsymbol{R}_i}\text{中},I_{e_{hj}}=0\end{cases} \quad (6)$$

$$V_{\boldsymbol{RS}_ih}=\begin{cases}1 & \sum_{i=1}^{n}W_{\boldsymbol{RSI}_ihj}/n\geqslant 0.6\\ 0 & \sum_{i=1}^{n}W_{\boldsymbol{RSI}_ihj}/n<0.6\end{cases} \quad (7)$$

式(7)表示结果项$\boldsymbol{R}_i$对应于类别S_h中所有特征中归属权重不为0的特征所占比例大于某一个阈值时将结果项$\boldsymbol{R}_i$归入类别S_h，本文中阈值是0.6.

6 算法分析

本节主要从算法复杂度对本算法进行分析.本节采用与第2节相同的测试数据，共1003个查询请求，具体统计数据如表2所示.

表2 实验数据统计列表

统计项	数值
单个结果项含有关键词平均数	27
单个结果项含有词频平均数	35
一次搜索任务中，所有结果含有关键词平均数	978
单个结果含有聚类特征的平均数	17

在传统方法中，为了进行聚类，每个样本需要对应含有大量特征的一维向量，例如本文中，其平均值为978，即“一次搜索任务中，所有结果含有关键词的平均数”；本算法力求实时性，所以对算法在时间和空间上都进行了优化，关键是对聚类特征进行了优化，抽取出现频率最高的关键词(占关键词总数的25%)作为特征进行计算，降低了算法复杂度.

7 结束语

KFC算法是对搜索引擎结果聚类的一次有益尝试，同传统聚类方法(如KNN[6])相比，该算法复杂度较低，聚类主题明确，效果较好，并且事先不需要给出初始聚类数，并能在聚类过程中自动调节新特征进入某一类时的权值阈值.

如何使聚类最大程度独立于搜索结果的语料情况、如何将语义引入到对搜索引擎的结果聚类中，并利用先验的知识库来指导搜索引擎结果聚类等，这些问题都值得进一步研究，使得聚类更准确、合理.

参考文献(References)

[1] Yitong Wang, Masaru Kitsuregawa. Use link-based clustering to improve web search results[C]//Proceedings of the 2nd International Conference on Web Information Systems Engineering. Washington: WISE, 2001: 119－128

[2] Zeng Huajun, He Qicai, Che Zheng, et al. Learning to cluster web search results[C]// Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval. New York: ACM Press, 2004: 210－217

[3] Andreas Hotho, Alexander Maedche, Steffen Staab. Ontology-based text document clustering [C]// Klopotek MA, Wierzchon ST, Trojanowski K. Proc of the Conf on Intelligent Information Systems. Zakopane: Springer-Verlag, 2003

[4] Wang Po-Hsiang, Wang Jung-Ying, Lee Hahn-Ming. Queryfind: search ranking based on users' feedback and expert's agreement[C]//IEEE International Conference on e-Technology, e-Commerce, and e-Service. [S. l.]: IEEE, 2004: 299－304

[5] 耿玉良，陈家琪，王咏梅. 中文Web检索中聚类算法的改进[J]. 计算机工程与设计，2005，26(10)：2685－2687
Geng Yuliang, Chen Jiaqi, Wang Yongmei. Improvement of clustering algorithm in chinese web retrieva [J]. Computer Engineering and Design, 2005, 26(10): 2685－2687(in Chinese)

[6] 姚莉秀，杨杰，叶晨洲，等. 用于特征筛选的最近邻(KNN)法[J]. 计算机与应用化学，2001，18(2)：135－138
Yao Lixiu, Yang Jie, Ye Chenzhou, et al. K nearest neighbor(KNN) method used in feature selection [J]. Computer and Applied Chemistry, 2001, 18(2): 135－138(in Chinese)

(责任编辑：张 婧)