基于相似度线性加权方法的检索结果聚类研究*

刘海波，郑德权，赵铁军
教育部-微软语言语音重点实验室，哈尔滨工业大学，哈尔滨 150001
E-mail: hbliu@mtlab.hit.edu.cn

摘　要：对检索结果的聚类能够便于用户在大量搜索结果中快速找到需要的信息，传统文本聚类技术在检索结果聚类上取得的效果并不好。Lingo 算法采用 LSI(潜在语义索引)对检索结果进行聚类，其首先生成候选标签，然后分配文档，形成聚类。本文提出一种在 Lingo 算法的基础上，融合 HowNet 语义相似度和余弦相似度线性加权的 Single-Pass 改进方法对聚类进行融合和簇再发现，并提取簇标签。该方法在聚类的纯度和 F 值方面均取得了较好的实验结果。

关键词：文本聚类；信息检索；Lingo 算法；语义相似度；余弦相似度

Study on the Retrival Results Clustering Based on Linear Weighting Method of Similarity

Liu Haibo, Zheng Dequan, Zhao Tiejun
MOE-MS Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, Harbin 150001
E-mail: hbliu@mtlab.hit.edu.cn

Abstract: The retrieval results clustering can facilitate the users in finding the needed from massive information. But the effect of the traditional text clustering has been verified no good. Lingo Algorithm, which adopts LSI (Latent Semantic Indexing) for clustering, generates candidate labels first, then distributes the documents, and form the clusters finally. On the basis of Lingo Algorithm, this paper presents a linear weighted method of Single-Pass improvement, which integrates HowNet semantic similarity and cosine similarity, fuses and rediscovers clusters, and extracting the cluster labels. The experiments have showed it has good performance in purity and F-measure of clusters.

Keywords: text clustering; information retrieval; lingo algorithm; semantic similarity; cosine similarity

1　引言

随着互联网的飞速发展，Web 上的信息呈爆炸式的速度增长，信息检索能满足用户在大规模信息中发现感兴趣的信息。但是往往搜索引擎给出的信息很庞杂，这就需要对检索结果进一步处理，其常用的方法就是检索结果聚类。Lingo 算法是一种高效的检索结果聚类算法，其主要基于矩阵奇异值分解生成聚类标签，然后经过相似度计算来分配文档。

本文在 Lingo 算法的基础上，提出了一种结合 HowNet 语义相似度和向量余弦相似度的线性加权的聚类标签融合方法。

2　相关工作

文本聚类是在没有学习的情况下将文档集划分为多个不同簇，每个簇内文档相似度高，簇间相似度低。针对检索结果的聚类是对搜索引擎返回与检索词相关的文档列表进行聚集，其只能通过文档标题、摘要等少量信息来进行，以满足实时性需求。检索结果聚类始于 Scatter-Gather 系统[3]，其采用了分馏算法。典型的检索结果聚类算法有：STC(Suffix Tree Clustering)[2]，后缀树算法，首先将文档构造成后缀树，并利用后缀树找到基类簇，即最大短语束，然后合并基类簇。

本文将基于描述优先的 Lingo 算法和启发式聚类算法 Single-Pass 相结合，采用基于语义相似

* 本文工作获得国家自然科学基金项目(No. 61073130)及哈尔滨工业大学科研创新基金项目(No. HIT.NSRIF.2009072)资助。

度和余弦相似度加权的 Single-Pass 改进方法对 Lingo 聚类进行融合。Single-Pass 方法对于文档次序的依赖较大，对于同一文档集，按不同次序聚类会出现不同结果。但对于检索结果聚类，由于检索的信息已具有次序信息，该方法具有较好的适用性。

3 Lingo 算法

传统聚类算法通常是首先生成类内容，然后根据内容生成类标签。因此会产生很多无意义标签，同时受噪声信息的影响也比较大。本文采用后缀数组获取常用短语，采用词-文档矩阵的奇异值分解（SVD）获取抽象概念，并通过短语匹配获取聚类标签，最后采用 VSM 模型来对聚类划分簇内容[4]。其主要分以下几个步骤：文档预处理、特征抽取、归纳标签、聚类内容发现及得分和排序。

在文档预处理阶段主要对检索结果返回的文档进行预处理，主要包括：噪声信息过滤，分词，数据格式化，停用词标识。

特征抽取阶段主要是发现可理解的词或短语，其可认为是类标签的候选。特征抽取基于以下几点：(1) 出现次数超过一定阈值；(2) 短语在一个句子内出现，并不跨越分隔符；(3) 短语完整性；(4) 不以停用词开始或结尾。首先对文档集中文档进行分词，并连接分词后的文档集[1]。根据文档集生成的后缀数组构建最长公共前缀（LCP）数组，并发现右完全短语（RCP）、左完全短语（LCP），将其均按字母序排列，在左完全短语和右完全短语中发现完全短语集。依据上述 4 个条件，对短语集进行过滤，过滤后的短语集即作为候选。

类标签的归纳是基于词文档矩阵的 SVD 分解的，主要分为四个阶段：

（1）构建词文档矩阵

将文档集采用 TF-IDF 权重构造成词-文档矩阵。词标识为停用词及词频低于阈值的词不进行处理，这样可降低维度。在 VSM 下，文档被表示为一个多维的向量，向量的每个维度表示词和该文档的联系。$TF(i, j)$表示词i在文档j中出现的频率，$IDF(i, j)$表示词i逆文档频率。

（2）抽象概念发现

对词文档矩阵 A 进行 SVD 分解，并计算 A 的 k 秩近似矩阵[5]。经过奇异值分解，矩阵 A 可以表示为：$A = U \Sigma V^T$。U 表示 A 的左奇异向量，V 表示 A 的右奇异向量。而 U 中每个列向量可表征一个抽象概念。

（3）短语匹配

将特征抽取出来的短语构建成一个矩阵 P，并将经过奇异值分解得到的抽象概念矩阵相乘得到 M 矩阵。从 M 矩阵中我们得到候选聚类标签，这些标签能最大程度地匹配抽象概念，同时去除无效短语。

（4）候选标签修剪

针对候选标签进行相似度计算，将相似度超过阈值的选择其中得分高的，去除其他的候选。候选标签间的相似度采用向量余弦相似度来计算：

$$\cos\theta_j = \frac{a_j^T q}{\| a_j \| \| q \|} = \frac{\sum_i a_{ij} q_i}{\sqrt{\sum_i a_{ij}^2} \sqrt{\sum_i q_i^2}}$$

其中，a_j^T 表示 VSM 模型中文档向量，而 q 表示标签向量。

类内容发现阶段主要按照上面生成的聚类标签将文档划分到类中。将生成的类标签即短语与文档采用向量余弦相似度进行计算，若超过阈值，则分配到该类。

在生成聚类标签时，聚类得分计算方法如下：

$$\text{CScore} = \text{LScore} * \text{CCount}$$

其中，CScore 表示聚类得分，LScore 表示标签得分，即在短语匹配时生成的候选标签得分，CCount 表示类大小，指类中文档数量。

4 聚类融合方法

经过 Lingo 算法得到的聚类，其仅仅依靠相似度计算来分配类别，并不能将具有相似语义信息的文档聚集到一类中。而且聚类内容存在重叠，需要进一步的融合。本文提出基于 HowNet 语义相似度和向量余弦相似度加权的 Single-Pass 聚类方法，该方法有两个基本假设：

假设 1，上述算法得到的聚类是有效地，即聚类标签是合理的、聚类内部相似性高；

假设 2，聚类的排序是合理的。

4.1 基于 HowNet 的语义相似度计算

HowNet（知网）是一个以汉语和英语的词语所代表的概念为描述对象，以揭示概念与概念之间以及概念所具有的属性之间的关系为基本内容的常识知识库。其对于一个词的语义采用多维的知识表示形式。在 HowNet 空间中，词用概念来描述，一个词可以表达为几个概念，而概念用义原描述。

对于两个词语W_1和W_2，如果W_1有 n 个义项（概念）：S_{11}、S_{12}、…、S_{1n}，W_2有 m 个义项（概念）：S_{21}、S_{22}、…、S_{2m}，规定W_1和W_2的相似度为其所含概念的相似度的最大值，即：

$$Sim(W_1, W_2) = \max_{i=1\ldots n, j=1\ldots m} Sim(S_{1i}, S_{2j})$$

由于所有概念都归结于用义原来表示，所以义原的相似度计算是概念相似度计算的基础。所有的义原根据上下位关系构成一个树状的义原层次体系，可以采用语义距离计算相似度。假设两个义原在这个层次体系中的路径距离为d，可定义两个义原之间的相似度为：

$$Sim(p_1, p_2) = \frac{\alpha}{d + \alpha}$$

其中，p_1、p_2表示两个义原，α为可调参数。

概念的相似度可以由义原相似度加权计算得到[6]。句子的相似度定义如下：

假设句子 A 和 B，A 分词和预处理后的词序列$A(A_1, A_2, \ldots, A_m)$，B 分词和预处理后的词序列$B(B_1, B_2, \ldots, B_n)$，词$A_i (1 \le i \le m)$和$B_j (1 \le j \le n)$的相似度可以用$sim(A_i, B_j)$表示，则句子的相似度定义为

$$s(A, B) = \left(\frac{\sum_{i=1}^{m} a_i}{m} + \frac{\sum_{j=1}^{n} b_j}{n} \right) \Big/ 2$$

其中，$a_i = \max(sim(A_i, B_1), sim(A_i, B_2), \ldots, sim(A_i, B_n))$，$b_j = \max(sim(A_1, B_j), sim(A_2, B_j), \ldots, sim(A_m, B_j))$

4.2 相似度线性加权的 Single-Pass 融合方法

将基于上述方法的聚类结果中每类看成是在相似度很高的部分文档集合，其构成的类别是有效的，类别标签能真实反映该类的信息。

将每一个类中文档连接为一篇文档，和没能划分类别的文档构成新的文档集。在此文档集上，采用语义相似度和余弦相似度线性加权的方法进行类别融合，处理过程如算法 1 所示。

算法 1：相似度线性加权的 Single-Pass 融合算法

（1）将上述所得每类中文档连接为一篇文档，构建文档集，并构建 TFIDF 矩阵；

（2）构建类集合为空；

（3）从第一篇文档开始，计算该文档与类集合中类别的余弦相似度 sim_{cos} 和语义相似度 sim_{HowNet}，计算综合相似度，方法如下：

$$sim(D,C) = \alpha\, sim_{HowNet} + (1-\alpha) sim_{\cos}$$

其中，α 为 0 到 1 之间的加权系数。

a) 若上述计算得到的相似度超过定义的阈值 β，则认为该文档属于该类，将文档合并入该类，并重计算类中心点。计算方法如下：

若该类中已含有该文档，则保持不变；

若类中没有该文档，连接文档，并采用均值重新计算 TFIDF 权重。

b) 若没有超过阈值，则由该文档形成新类别。

（4）迭代进行计算，直至文档集分配结束；

（5）将新类与原始类别匹配，定义类别重合度 δ [8]，其计算方法如下：

$$\delta = \frac{|C_o \cap C_n|}{|C_o \cup C_n|}$$

其中，C_o 表示原始类别，C_n 表示新类，δ 等于 C_o 和 C_n 中共同的文档数量比上 C_o 和 C_n 中所有文档数量。

a) 如果超过了给定的阈值 γ，则采用原始类别标签作为新类标签。

b) 如果新类与多个原始类别的重合度超过阈值，则取重合度最大的作为标签。

c) 如果没有超过阈值，则采用基于后缀数组的方法抽取完全短语，并按前面所述规则过滤。在此基础上添加过滤规则，即：类别中包含该短语的文档数应该超过该类中文档总数的一半。在抽取的短语集合中取 TF-IDF 权重最大的短语作为标签。

d) 若没有抽取到短语，则去除该类。

（6）类别按照生产的顺序进行排序。

5 实验及结果分析

本文采用从互联网搜索引擎检索出的真实数据进行实验，从算法的时间复杂度、类标签质量、聚类结果的纯度和 F 值（F-Measure）等方面进行评价。

5.1 评价标准

对聚类结果的评价经常采用两种指标：纯度和 F 值[7]。F 值（F-measure）是采用信息检索中经典的准确率（precision）和召回率（recall）的组合来进行聚类评价。对于聚类 j 及与此相关的正确分类 i 的准确率和召回率定义为：

$$P = precision(i,j) = N_{ij} / N_i \qquad R = recall(i,j) = N_{ij} / N_j$$

其中，N_{ij} 表示聚类 j 中分类 i 的数量，N_i 表示分类 i 中所有文档数量，N_j 表示聚类 j 中所有文档数量。则分类 i 相应 F 值定义为：

$$F(i,j) = 2PR/(P+R)$$

对分类 i 而言，哪个聚类的 F 值最大，就认为该聚类代表分类 i 的映射。整体聚类的 F 值可以由每个类别的 F 值加权得到：

$$F = \sum_i \frac{N_i}{N} \max\{F(i,j)\}$$

其中，N 表示文档总数，N_i 表示分类 i 中所有文档数量。

纯度是简单的聚类评价方法，是指正确聚类的文档数占文档总数的比例。

针对检索结果聚类，需要考虑到系统实时性的要求，本文也从时间复杂度对检索结果聚类进行了评价。

5.2 实验方法和结果分析

本文采用文本聚类和检索结果聚类两种不同的方法进行实验。在文本聚类时，通过爬虫在100多个综合站点中抓取信息，然后抽取出某个时间段内的1000条进行实验。基于检索结果的聚类，本文采用Bing搜索引擎，采用5个查询，对检索结果的前100条进行聚类。实验均采用基于语义相似度和余弦相似度加权的方法和未作融合的方法进行比较，以及和仅用余弦相似度、仅用语义相似度融合的方法进行比较。实验结果均采用人工标注的方式进行统计和分析。

（1）文本聚类有效性分析

对文档集我们分别采用Lingo算法、余弦相似度融合方法、语义相似度融合方法和相似度加权融合方法进行聚类。首先，我们对文档集进行人工分类，并以此作为评价标准。在此基础上，对聚类内容进行有效性评价，从而得到文本聚类有效性展示图。

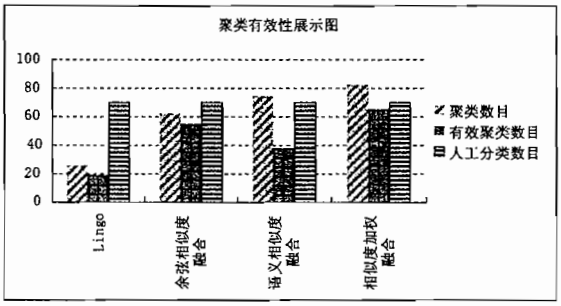

图1 文本聚类有效性展示图

从图1中可以看出，基于余弦相似度融合和基于相似度加权的方法能较大提高聚类有效性，且聚类的覆盖率也有了较大提升。基于语义相似度融合的方法能发现较多类别，但是有效类别的比例较小。分析原因主要是通过语义的扩展，该方法能发现更多类别，但不能反应相似词语的重要程度，噪音信息很多。从图中也可以看出，基于语义相似度和余弦相似度加权的方法能发现更多类别，这表明结合语义相似度和余弦相似度融合的方法既对类别进行了语义扩展，同时也包含了词的权重信息，聚类的有效性有了较大提高。

（2）检索结果聚类

我们采用5个查询，分别为：信息检索、文本聚类、福岛地震、利比亚局势、苹果。通过实验发现在加权系数α为0.3时其能取得较好的结果。

首先，我们对聚类时间做了比较，结果如表1。

表1 聚类时间比较

查询条件	运行时间（ms）			
	Lingo 算法	*余弦相似度融合*	*语义相似度融合*	*相似度加权融合*
信息检索	5230	6580	5793	7830
文本聚类	4344	6793	7007	8920
福岛地震	4749	6637	7575	8632
利比亚局势	4101	6641	6847	8784
苹果	4166	7080	6928	8797

表1中运行时间是指从启动到输出聚类结果所用时间，其包括从搜索引擎返回结果的过程。这和文献5中提到的运行速度相比慢，分析原因有以下两点：1) 网络延时比较高；2) 装载词典耗时较多。从运行时间来看，基于相似度加权融合的方法比Lingo算法和仅用余弦相似度或语义相似度融合的方法耗时多。

其次，我们对聚类结果进行人工标注，给出标准类别，经分析得到聚类的 F 值和纯度图。

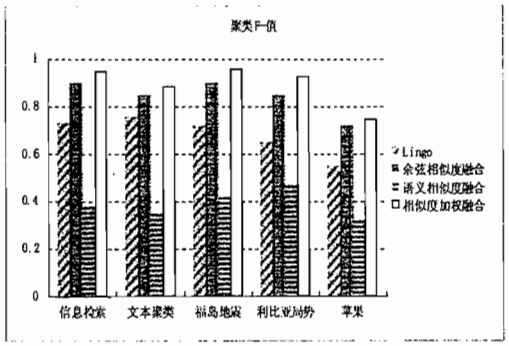

图 2　聚类算法 F-值比较

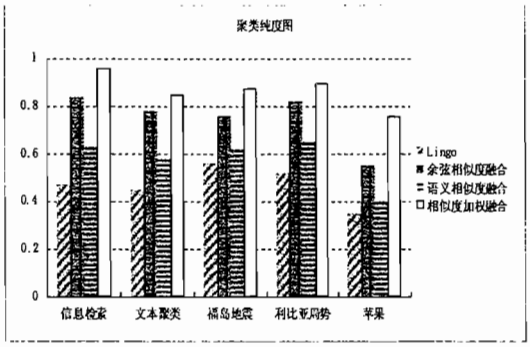

图 3　聚类算法纯度比较

从图 2 和图 3 可以看出，基于相似度加权融合的方法比基于余弦相似度或语义相似度融合的方法和 Lingo 算法都有一定提高。基于语义相似度的融合方法得出的 F-值较低，分析原因为仅仅采用语义相似度进行融合，类别准确率较低，且存在部分重叠。基于相似度加权融合的方法在发现新类的同时也会在产生语义上的一些噪音，这也是下一步需要研究的。

6　结论

本文首先在Lingo聚类的基础上，将HowNet语义相似度和余弦相似度加权来计算综合相似度，采用单次遍历的方法对聚类进行融合和再聚类。实验表明，该方法具有较好的效果，且能满足检索结果聚类的要求。

针对检索结果的聚类，我们下一步的工作主要从以下两个方面出发：第一，在特征抽取阶段，对抽取的特征进行语义融合，以保证语义信息更明显，也能更好地去除噪声。第二，有效的减少聚类时间。

参 考 文 献

[1] Dell Zhang and YishengDong, Semantic, Hierarchical, Online Clustering of Web Search Results. Accepted by 3rd International Workshop on Web information and data management, Atlanta, Georgia.

[2] Oren E. Zamir. Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. Doctoral Dissertation, University of Washington, 1999.

[3] D. Cutting, D. Kargcr, J. Pedcrscn, J. W. Tukcy.Scattcr/Gathcr: A Clustcr-bascd Approach to Browsing Largc Documcnt Collections. In Proceedings of the 15th Annual International ACM/SIGIR Conference, Copenhagen, 1992.

[4] Stanis law Osi'nski, An Algorithm for Clustering of Web Search Results. Submitted in partial fulfillment of the requirements for the degree of Master of Science, Poznań University of Technology, Poland, 2003.

[5] Stanis law Osi'nski, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on Singular Value Decomposition. Submitted to Intelligent Information Systems Conference 2004, Zakopane, Poland, 2003.

[6] 刘群，李素建. 基于《知网》的词汇语义相似度计算. 第三届汉语词汇语义学研讨会，台北，2002.

[7] 吴启明，易云飞，文本聚类综述. 河池学院学报，2008，28(2): 86-91.

[8] 刘之涛，陈清才，孟宪军，王晓龙. 基于特征短语的网页在线聚类方法. 第四届全国信息检索与内容安全学术会议. 2008.