第 33 卷 第 8 期 计 算 机 工 程 2007 年 4 月
Vol.33 No.8 Computer Engineering April 2007
·博士论文· 文章编号：1000—3428(2007)08—0051—03 文献标识码：A 中图分类号：TP391

一种基于语言概念空间聚类的信息检索方法

吴 晨[1,2]，张 全[2]
(1. 中国科学院研究生院，北京 100039；2. 中国科学院声学研究所，北京 100080)

摘 要：提出了一种以语言概念空间中的概念为聚类对象的信息检索方法以及适合于该方法的聚类算法。该聚类算法通过曲线拟合技术来实现文本的自动阈值确定和聚类划分，并最终通过聚类间的迭代和结果修正来完成整个聚类过程。概念的引入为解决词语的同义、多义问题提供了有力保障。实验表明，采用该方法的信息检索系统，与 Jelinek-Mercer、k-means 模型相比有较高的准确率和召回率，效果理想。
关键词：信息检索；语言概念空间；聚类；自动阈值下的聚类划分

An Information Retrieval Method Based on Language Concept Space Using Clustering Method

WU Chen[1,2], ZHANG Quan[2]
(1. Graduate School, Chinese Academy of Sciences, Beijing 100039; 2. Institute of Acoustics, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 An information retrieval model based on language concept space and a clustering method which serves the IR model is propsed. The clustering method uses curve-fitting to implement the text clustering by auto threshold-detection means, and complete the whole clustering process through result revising phase. The use of word concept can reduce the word sense ambiguity as drastically as possible when processing the text. The experiments indicate that the method presented in this paper has good performance. Compared with Jelinek-Mercer smoothing model and k-means model, the precision and the recall of the system are higher to a certain degree.
【Key words】 Information retrieval; Language concept space; Clustering; Auto threshold-detection and classification

从 20 世纪 70 年代开始，聚类在信息检索、情报学以及模式识别等领域中的应用就引起了广泛的关注。本文的主要任务一方面放在聚类方法的研究上，另一方面则放在语言概念空间的探索上。目前所有的聚类方法都无一例外地以文本中词语作为处理的主体，本文试图从概念的角度出发，结合我们在概念空间方面取得的研究成果[3,4]，来处理信息检索中的问题。作者认为，要使自然语言处理得到长足的进步，必须将对语言的处理深入到语义理解的层面，而基于概念空间的方法是一个非常好的选择。同时，由于对概念的聚类是一项尝试性较强的工作，我们尽可能地采用自动确定阈值及划分的方法来避免缺少经验数据的问题。

1 概念空间对照知识的建立

对于语言概念空间的研究是 20 世纪末兴起的，主要代表包括HNC(概念层次网络)理论[1,2]、WordNet理论[3]、知网理论[4]。后两者对于概念的研究范畴稍窄，主要集中在词汇层面。本文中用到的语言概念空间的知识主要以HNC框架为基础，集中在词语和句子的概念分析层面上。

表 1 词语知识库关键信息

项名称	说明
词语	记录词语的原型
义项数	所包含的义项数量
概念符号(义项)	描述义项的概念符号，根据“义项数”可能有多项
所属句子结构	不同的义项会对应不同的句子结构，句子结构也是一个概念层面的符号表达式[3]，用来描述句子的语义结构

首先需要建立一个词语到概念的知识对照表(以下简称“词语知识库”)，对照表的功能是实现词语到概念间的映射和概念到词语间的反映射，建立对照表的依据是HNC概念节点表[3]。我们所建立的词语知识库中包含的关键信息如表 1 所示。

2 基于概念聚类的信息检索方法

2.1 CCIR 方法的基本原理

在概念聚类的信息检索方法(以下称 CCIR 方法)中，引入概念的思想，文档不是用词语，而是用概念符号来作为表示的对象，然后采用迭代处理的聚类方法进行计算。这种方法不需要预先计算相似矩阵。

引入概念的最大好处在于概念不存在语义模糊，一个概念符号对应一种确定的语义，这可以从根本上解决同义词、多义词问题。于是所有的文档就可以表示成一个以概念为内容的巨大的稀疏矩阵。与此同时，带来的问题就是如何进行词语概念的认定，如果概念能够成功地认定，那么信息检索系统的任务就可以看作是判断哪个文档所代表的概念与检索概念最接近的问题，这一判断问题交由聚类算法去完成。考虑到基于概念的聚类算法可供参考的经验数据很少，尤其是经验阈值、聚类划分数量的确定，于是在 CCIR 方法中我们

基金项目：国家“973”计划基金资助项目“自然语言理解的交互引擎研究”(2004CB318104)；中科院声学研究所知识创新工程项目“HNC语言知识处理理论及技术”
作者简介：吴 晨(1979－)，男，博士生，主研方向：自然语言理解；张 全，研究员、博导
收稿日期：2006-05-20 **E-mail：**wuchen@mail.ioa.ac.cn

提出了一种自动获取阈值的方法，具有一般意义。这种方法以某一文本与其它文本间的语义距离为 y 轴，以根据距离值递增排序的文档号为 x 轴所构成的曲线为依据，通过计算曲线的拐点来计算阈值，并根据阈值自动形成聚类。最后通过基于概念的查询条件与概念聚类之间的平滑模型匹配来形成查询结果。

2.2 词语概念的认定

词语概念的认定通过HNC句类分析技术[3]实现，句类分析可以给出具体的语言概念层面上的语义结构以及各组成成分之间的语义关系。这里给出一个例子，可参见文献[3]。

例 1 国际奥委会 ‖ 公布 ‖ 评估 [报告]

由句类分析给出的它的语义结构是

T3Y30*21J = TA(国际奥委会：fpea339)+T3Y30(公布：vc331)+YC(评估：v841&报告：gwc239ea2)

其中，T3Y30*21J是句子所属的句类，它表示这句话的语义框架结构；TA、T3Y30、YC表示这个框架的组成内容，即构成语义块[1]。词语后面紧跟的字符串是词语在此句中的语义的概念表示，这是后续处理所需要的。由于目前还无法客观地分析句类分析获得的语义结构中各个语义块中概念所具有的权重值，因此概念在语义结构中所处的位置，对其权重的影响在CCIR中没有考虑。

2.3 聚类的生成

寻找适合对概念进行聚类的生成算法是本文的一个研究重点，k-means 聚类算法由于其在时间复杂度上的优势成为首要考虑的算法，但是它需要主观指定 k 个文档来作为初始聚类的凝聚点，如果遇上孤立点算法还会表现得非常的敏感，这是 CCIR 方法中不期望的。我们希望算法能够通过数据的特点自动地划分初始聚类，这就是本节将讨论的问题。本节将从文本距离度量标准、聚类阈值的确定、聚类的确定 3 个部分进行描述。

2.3.1 文本距离度量标准

文本聚类算法中很重要的一个度量标准是文本和文本之间、文本和聚类之间、聚类和聚类之间的距离度量标准。CCIR 方法中采用了 Kullback-Liebler 算法。

$$KL(d,c)=\sum_{w_i\in d}\frac{n(w_i,d)}{|d|}\log\frac{n(w_i,d)/|d|}{n(w_i,c)/|c|}$$

其中，$KL(d,c)$ 表示文档(类)d 和文档(类)c 之间的语义距离，显然有 $KL(d,d)=0$。$n(w_i,x)$ 为概念 w_i 在文档(类) x 中出现的次数。$|x|$ 为文档(类) x 中概念出现的总数。

2.3.2 聚类阈值的确定

要自动地划分聚类，必须找到划分的标准，我们希望通过文本相似度上的阈值来确定聚类划分的标准。CCIR 方法的基本思想为：对于一个文本，与它相似的文本和与它不相似的文本，从总体上看，文本距离会出现密度上的跳跃，这些跳跃点就是我们要寻找的阈值。如果将这一文本与其它文本之间的距离值递增排序，如图 1 所示，则可以通过拟合该函数，再根据函数的拐点来确定相应的阈值。

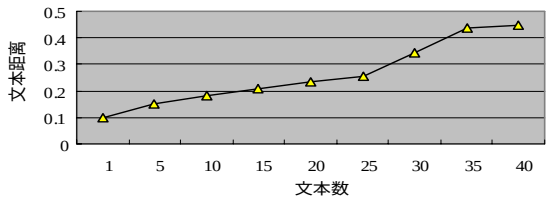

图 1 文本距离曲线

CCIR 采用最小二乘法的思想来对曲线进行拟合。问题转化为求 $f(x)$，使得 $\delta=\sum_{i=1}^{n}\delta_i^2=\sum_{i=1}^{n}[f(x_i)-y_i]^2$ 达到最小。其中 δ_i 为点 (x_i,y_i) 与曲线 $y=f(x)$ 在点 x_i 处的距离。假设 $f(x)=c_0+c_1x^1+c_2x^2+\ldots+c_mx^m$，现在要确定系数 c_0、c_1、c_2、…、c_m，使 δ 达到最小。为此，将 $f(x)$ 的表达式带入 δ 中，δ 就是 c_0、c_1、c_2、…、c_m 的函数，求 δ 的极小值，可令 δ 对 c_j 的偏导数等于 0，得到 m+1 个方程组，从方程组中求解出 c_j 即可。根据实际拟合效果发现用三阶多项式来对曲线进行拟合最为合适。

根据拟合得到的方程 $f(x)$，求其二阶导数，并且使得 $f''(x)=0$，得到的 x 就是曲线拐点，默认取第 1 个拐点，也就是 x 最小的点所对应的 y 作为所要求的阈值。

2.3.3 聚类的确定

求得阈值以后就可以进行聚类生成了，确定聚类的基本思想是：

首先随机地选取某一个文档，计算该文档与其它所有文档间的语义距离，根据 2.3.2 节所述方法求得阈值，语义距离在阈值之内的文档被归为一个初始类，然后以与随机选取的文档语义距离最远的文档为核心，计算它与其它未被聚类的文档之间的语义距离，并求得阈值，以这一阈值为依据，划分第 2 个初始类，如此进行下去，直到所有文档均被划分到某个类。通过一次聚类运算得到的初始聚类集所包含的类的数量可能较高，我们对得到的初始聚类集再进行聚类生成，方法与生成初始类的方法相同，只是此时以初始类为对象，计算这个类与其它类之间的距离，获取阈值并进行合并，如此迭代直到生成的聚类的数量在某个可接受范围之内或者聚类数量无明显提高。聚类生成的最后一个阶段任务是进行结果的校正，以前两阶段获取的聚类为依据，遍历文档集中的每一个文档，计算该文档与所有聚类之间的距离，如果该文档与原聚类间的距离最近，不作改动，否则将文档转移到与此文档语义距离最近的聚类。在聚类生成的过程中，必须考虑孤立点的问题。系统通过设定值 MaxKL 来判定，当文档(类)与其它文档(类)之间的语义距离的最小值都大于 MaxKL 时，断定该文档(类)为孤立点。

聚类生成的具体步骤如下所述。

S1：从文档集合 S 中任意挑选一个文档 d 作为初始文档。

S2：根据 2.3.1 节距离公式计算文档 d 与文档集合 S 中所有文档 c 之间的语义距离，获取语义距离集合 $KL_d=\{kl_{c1},kl_{c2},kl_{c3},\ldots,kl_{cm}\}$。

S3：如果 $\min\{kl_{c1},kl_{c2},kl_{c3},\ldots,kl_{cm}\}>MAXKL$，划分初始聚类 P，P={d}，转 S5。

S4：根据 $KL_d=\{kl_{c1},kl_{c2},kl_{c3},\ldots,kl_{cm}\}$，通过 2.3.2 节描述的阈值计算方法确定阈值 $kl_{threshold}$，并根据阈值划分一个以文档 d 为中心的初始聚类 P，$P=\{cn\,|\,kl_{cn}<=kl_{threshold},cn\in S\}+\{d\}$。

S5：从文档集合 S 中剔除被分类的文档，即 S=S-P。

S6：判断 S 中是否还有文档，如果没有，转 S8。

S7：取 kl= $\max\{kl_{c1},kl_{c2},kl_{c3},\ldots,kl_{cm}\}$ 所对应的文档为 d，重复 S2~S7。

S8：判断 S1~S7 计算后获得聚类的数量是否在设定的范围之内，如果是，转 S17。

S9：随机地从聚类集 SC 中选定一个聚类 P。

S10：计算 P 与 SC 中其它聚类之间的距离，获取语义距离集合 $KL_p=\{kl_{p1},kl_{p2},kl_{p3},\ldots,kl_{pm}\}$。

S11：如果 $\min\{kl_{p1},kl_{p2},kl_{p3},\ldots,kl_{pm}\}>MAXKL$，保留原聚类，

转 S13。

S12：根据 $KL_p=\{kl_{p1},kl_{p2},kl_{p3},\ldots,kl_{pm}\}$，通过 2.3.2 节描述的阈值计算方法确定阈值 $kl_{threshold}$，并根据阈值合并生成新的聚类 P，$P=\{cn\mid kl_{pi}<=kl_{threshold},cn\in Pi\}+P$。

S13：从 SC 中剔除被处理的聚类。

S14：判断 SC 中是否还有未处理聚类，如果没有，转 S16。

S15：取 kl= $\max\{kl_{p1},kl_{p2},kl_{p3},\ldots,kl_{pm}\}$ 所对应的聚类 P，重复 S10~S15。

S16：判断新生成的聚类数量与原聚类数量有否明显减少，如有重复 S8~S16。

S17：遍历所有文档 d，计算其与所划分的聚类之间的语义距离，取 $kl_{pi}=\min\{kl_{p1},kl_{p2},kl_{p3},\ldots,kl_{pk}\}$，如果该文档已经属于 Pi，不作处理，否则将该文档重新划分至 Pi。

S18：聚类生成处理结束。

以上步骤可以划分为 3 个阶段：S1~S7 为初始聚类生成过程；S8~S16 聚类迭代合并过程；S17 为聚类结果校正过程。S2~S7 与 S10~S15 处理原理相同，实际编写算法时可合并。

2.4 查询条件与文档之间的匹配

进行查询条件与文档之间相似度度量时，CCIR 希望充分发挥按照 2.3 节方法产生的聚类分布均匀、类内聚性高的特点，参考 Dirichlet smoothing 模型，设计了非标准的 Dirichlet smoothing 模型。它通过个体、局部、全局 3 部分出现查询条件的极大似然概率间的差值来计算文档 d_i 下出现查询条件的概率。同样，CCIR 采用词语概念作为处理的对象，由于在少量关键词的情况下确定用户输入词语所代表概念的偶然性非常大，我们并没有试图去确定词语的概念，而是根据词语表达某一概念的可能性来加权计算。这一数值通过计算文档集中出现概念的次数与表示该概念的词语的次数之间的比值来确定。

CCIR 方法中设计的基于概念的非标准 Dirichlet smoothing 模型计算公式如下：

$$P_{Dir}(Q\mid d_i)=\prod_{q_j\in Q}\sum_{n}P(q_j,w_{jn})\times\frac{n(w_{jn},d_i)+\mu\sum_{d_k}P(w_{jn}\mid d_k)P(d_k\mid S_{P1,P2,\ldots,Pm})}{\mid d_i\mid+\mu}$$

其中 $n(w_{jn},d_i)$ 为概念 w_{jn} 在文档 d_i 中出现的次数；$\mid d_i\mid$ 为文档(类)中概念出现的总数；$P(d_k\mid S_{P1,P2,\ldots,Pm})$ 为在聚类 $P1,P2,\ldots,Pm$ 分布下，出现文档 d_k 的可能性。

$$P(d_k\mid S_{P1,P2,\ldots,Pm})=\frac{1-kl_{Ps}}{\sum_{Pi=P1}^{Pm}(1-kl_{Pi})}$$

其中 Ps 为 d_k 所在的聚类；kl_{Pi} 为文档 d_k 到聚类 Ps 之间的距离。在 2.3.3 节的 S17 中已经计算过。

$P(w_{jn}\mid d_k)$ 为在文档 d_k 中出现关键概念 w_{jn} 的概率。

$$P(w_{jn}\mid d_k)=\frac{n(w_{jn},d_k)}{\mid d_k\mid}$$

$P(q_j\mid w_{jn})$ 为概念 w_{jn} 由词语 q_j 生成的概率。

$$P(q_j\mid w_{jn})=\frac{n(w_{jn},q_j)}{n(w_{jn},S)}$$

式中，$n(w_{jn},q_j)$ 为由 q_j 生成的 w_{jn} 的数量；$n(w_{jn},S)$ 为文档集 S 中 w_{jn} 的总数。μ 为 Dirichlet 参数。

3 方法测试

为了评测 CCIR 方法，对基于 CCIR 方法实现的信息检索系统 CCIRSys 进行了测试，并从乔治华盛顿大学获取了基于 Jelinek-Mercer smooth 模型、Bayesian 模型的信息检索测试系统，将这两个系统进行适合中文处理的修改以后，与 CCIRSys 进行了对比测试。

3.1 实验测试集

系统的测试数据为 TREC6 提供的中文信息检索(跨语种信息检索 E-C)测试数据集，测试集共包括 164 811 篇文章，这些文章全部来自于人民日报和新华网，共 170MB 生语料。测试主题为 TREC6 设置的 26 个新主题，这些主题都以中、英文两种形式存在，以便跨语种信息检索系统测试使用，我们只用了其中的中文部分。每个主题根据 TREC6 的统计平均有 114 相关中文文档。TREC6 并且提供了正确的检索结果集。根据 TREC6 的要求，在查询时返回前 1 000 篇文档作为返回结果。

3.2 实验结果

CCIR 方法中涉及了多项参数设置，在 CCIRSys 系统中根据模型特点和实际测试效果对参数进行了最终的设定。首先是 2.3.3 节算法的 S8 中，判断获得聚类的数量是否在设定的范围之内，范围设定的依据是文档总数，当文档数量大于 1 000 时，聚类的数量在 $10\log D_{number}$ 这样一个数量级；当文档数量小于等于 1 000 时，聚类数量大概在 $D_{number}/50$ 的数量级。其次是 2.3.3 节算法的 S16 中，判断新生成的聚类数量与原聚类数量有否明显减少，判断依据是聚类数量最高数字有否变化，如果没有变化则认为变化不明显。再有就是聚类生成算法中处理孤立点问题的阈值 MaxKL，将初始值设置为 0.85，但实际测试中表明，数值在 0.8~0.9 之间，算法都有相当好的孤立点容忍度，对最终结果影响极小。最后是对 Dirichlet 参数 μ 的确定，不同的查询条件受参数 μ 的影响程度不同，我们测试了 100、200、300、500、800、1 000、1 500 共 7 个数据。发现在 TREC6 测试集下，返回 1 000 个结果的条件下，平均正确率最高点出现在 $\mu=200$ 时，于是 CCIRSys 系统中的 μ 值设置为 200。

我们随后对 CCIRSys、Jelinek-Mercer、Bayesian 等 3 个系统进行了对比测试，对比的方法采用查准率(precession)-召回率(recall)对比图。对比图如图 2 所示。从试验数据中可以看出，CCIRSys 在召回率升高的情况下，查准率较其它两个模型要好很多，这主要是由于 CCIRSys 系统中概念的引入大大地削减了词语的同义、多义模糊，但同时也发现在这种方式下，常常发生用户检索到的文档中包含的词语是与用户输入同义的另一个词语，这是造成系统查准率损失的一个主要原因，有待进一步改善。系统性能提升的另一方面原因则是 CCIRSys 系统所采用的聚类算法可以很好地做到聚类划分的均匀性，为查询条件与文档之间的匹配服务。而 Bayesian 模型在召回率为 0.1~0.4 之间有较大的查准率抖动，这和我们测试时定义的聚类的数量以及随机抽取的样本凝聚点有很大关系，在测试时并没有试图找寻最优化的 Bayesian 初始聚类划分数量及 Jelinek-Mercer 模型中最优化的参数 λ 值，而是根据基于这些模型的聚类算法中常用的经验参数值来设置。

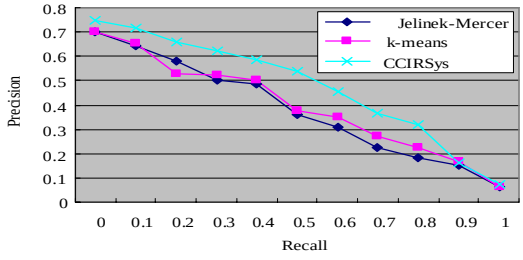

图 2 查准率-召回率对比 (下转第 56 页)