第 33 卷 第 2 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.2 Computer Engineering January 2007
·人工智能及识别技术· 文章编号：1000—3428(2007)02—0158—02 文献标识码：A 中图分类号：TP391

分布式信息检索的集合选择研究

张 刚[1,2]，郭 岩[1]，张 凯[1]
(1. 中国科学院计算技术研究所软件室，北京 100080；2. 中国科学院研究生院，北京 100039)

摘 要：集合选择是分布式信息检索中的重要问题，将集合选择问题转化为文档检索问题，尝试了多种文档检索方法来解决集合选择问题，并将各种方法的文档检索结果与集合选择结果进行了对比，通过与经典的集合选择算法 CORI 相比较，实验发现语言模型的集合选择方法能够取得令人满意的结果。

关键词：分布式信息检索；集合选择；语言模型

Research on Resource Selection in Distributed Information Retrieval

ZHANG Gang[1,2], GUO Yan[1], ZHANG Kai[1]
(1. Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080;
2. Graduate School of Chinese Academy of Sciences, Beijing 100039)

【Abstract】 Resource selection is one of the most important problems in distributed information retrieval. In this paper, resource selection is converted to document retrieval problem. Some document retrieval method is adopted to solve the resource selection problem. Each retrieval method is tested both on the document retrieval and resource selection problem. Compared to the well known resource selection method CORI, language model based retrieval method achieves a quite promising result.

【Key words】 Distributed information retrieval; Resource selection; Language model

Web信息的快速增长，给检索系统带来了巨大的挑战。据Hobbes' Internet Timeline统计，截止 2004 年 12 月，互联网上Web服务主机数已达到 56 923 737 台。面对这样大规模的信息检索问题，传统的集中式信息检索方法会遇到很多难以克服的困难[1]，而分布式信息检索通过将文档集合进行合理的划分，每次检索时利用集合选择算法找出最相关的数据集合进行检索，从而实现查询部分文档集合而给出很好的检索结果的效果，因此分布式信息检索是海量信息检索的有效解决方案。

分布式信息检索是信息检索的一个重要研究方向，研究的主要内容包含“集合划分”、“集合选择”、“单数据集合检索”、“结果合并”等几个部分。

集合选择的效果直接决定着最终检索结果的质量。在以往的研究中，CORI (Collection Retrieval Inference Network)算法[2]和gGlOSS(Generalized Glossary of Servers Server)算法[3]是最著名的 2 种。在Mikhail Sogrine等人的研究中对这 2 种集合选择算法进行了深入的比较，实验表明 CORI算法的结果要好于gGlOSS算法的结果[4]。

1 文档检索与集合选择问题的比较

文档集合由很多文档组成，如果把一个文档集合作为一个虚拟的文档，可以对这个虚拟文档建立索引，实现对文档集合的检索，进而将集合选择问题转化成一个文档检索问题，只是检索结果得到的不是一个真实的文档而是一个文档集合。鉴于集合选择问题被转化成一个文档检索问题，传统文档检索的方法就可以应用在集合选择问题上作为集合选择算法，事实上所谓的 CORI 算法就是文档检索中 INQUERY 算法在集合选择问题中的应用。但是文档集构成的虚拟文档与真实文档之间有很大差异，突出表现在以下几个方面：

(1)文档长度的不同

文档长度的不同是文档检索和集合选择很重要的不同点，在文档检索问题中，文档的长度为真实文档的长度，而在集合选择问题中，文档长度是虚拟文档的长度，也就是文档集合的长度，通常文档集合中包含大量的文档，其长度要远远大于真实文档的长度，而且虚拟文档的长度差异也远大于真实文档之间的长度差异，而很多检索方法和文档的长度有比较大的关系，因此文档长度的不同可能会对检索算法的效果有比较大的影响。

(2)文档内容的蕴涵不同

在文档检索问题中，一篇文档的内容往往比较单纯，不会出现很多的主题，而对于集合选择问题则不同，在一个集合中包含着诸多文档，每个文档都可能讨论不同的内容，因此文档集合构成的虚拟文档在内容上蕴涵更加丰富，一个虚拟文档中包含着多个主题，这也将对检索造成一定的影响。

(3)词语的分布特征不同

在文档检索问题中，真实文档较短，含有的词语相对较少，而在集合选择问题中，由多个文档构成的虚拟文档含有大量的词语，这样使虚拟文档的词语分布与真实文档的词语分布也有很大的不同，例如：通常查询的关键词不可能在每个真实文档中都出现，但每个关键词几乎在每一个虚拟文档中都会出现。

基金项目：国家“973”计划基金资助项目“大规模文本内容计算”(2004CB318109)

作者简介：张 刚(1977－)，男，助理研究员，主研方向：信息检索，自然语言处理；郭 岩，博士、助理研究员；张 凯，助理研究员

收稿日期：2006-01-11 **E-mail：**gangzhang@ict.ac.cn

2 经典检索方法介绍

信息检索经过几十年的发展，形成了一系列比较有效的信息检索方法，下面将对部分经典的文档检索算法进行简单的介绍。

2.1 tf.idf 系列检索方法

tf.idf 检索模型是最早被提出的比较有效的检索模型之一，在 tf.idf 模型中，tf 是关键词 t 在文档中出现的次数，它体现了“如果一个关键词在文档中出现得越多，说明这篇文档可能与查询越相关”的假设，而 idf 称为反文档频率，它与包含关键词 t 的文档个数成反比，idf 体现了“如果一个关键词在越多的文档中出现，那么这个关键词对于确定文档与查询是否相关的意义越小”的假设。在 tf.idf 方法中有很多不同方式计算 tf 值与权重规格化的形式。这里介绍其中的 2 种计算 tf 的方法，对于查询 q 的关键词 t 的权重表示如下：

$$w_{q,t} = tf_{q,t} * idf_t \tag{1}$$

其中 idf 是关键词 t 在文档集合中的 idf 值，该词在文档 d 中的权重表示为

$$w_{d,t} = tf_{d,t} * idf_t \tag{2}$$

如果查询 q 有 n 个关键词，则查询 q 与文档 d 的相似度计算方法如下：

$$Sim(d,q) = \sum_{t=1}^{n} w_{d,t} * w_{q,t} \tag{3}$$

这种 tf.idf 权重计算方法有时候不能取得满意的效果，这主要是因为长的文档往往具有更大的 tf 值，而这并不能说明长的文档就一定比短的文档与查询更相关，因此有一系列将 tf 进行规格化的方法，其中对数 tf 是比较常用的一种，通过对数变换可以将 tf 值压缩到一个比较小的范围内，从而减弱文档长度对于 tf 的过度影响。另外最常用的对权重进行规格化的方法就是“cosine normalization”，它用 tf*idf 得到的权重除以文档向量的欧几里德长度来进行规格化，即通常所说的向量夹角余弦的相似度表示方法，计算方法可以表示为

$$Sim(d,q) = \frac{\sum_{t=1}^{n} w_{d,t} \bullet w_{q,t}}{\sqrt{\sum_{t=1}^{n} (w_{d,t})^2 \bullet \sum_{t=1}^{n} (w_{q,t})^2}} \tag{4}$$

2.2 INQUERY 检索算法

INQUERY检索算法[6]是Turtle和Croft等人提出的一种基于贝叶斯网络的概率检索模型。在INQUERY方法中，为每个索引词、文档、用户查询建立一个随机变量，与文档d相联系的随机变量表示在检索过程中观察到这个文档的可能性；观察到一个文档之后，这个事件又造成“观察到这个文档中出现的索引词”这个事件有一个概率，赋给与索引词相联系的随机变量；与用户查询相联系的随机变量表示用户的信息需求得到满足的概率。具体的计算方法如下所示：

$$T = \frac{tf}{tf + 0.5 + 1.5 \times cw / avg\_cw} \tag{5}$$

$$I = \frac{\log(\frac{C + 0.5}{df})}{\log(C + 1.0)} \tag{6}$$

$$p(r_k \mid d) = b + (1 - b) \bullet T \bullet I \tag{7}$$

其中：tf是文档d中含有关键词r_k的个数；cw是文档d中关键词的数量；avg_cw是各个文档中关键词的平均数；C是文档集中文档的数量；df是含有关键词r_k的文档个数；b是最小置信因子，通常取值 0.4。

2.3 语言模型检索算法

语言模型的检索算法是 1998 年由Ponten等提出的一种新的基于概率的检索模型[7]。语言模型的检索算法对于查询和每个文档都建立了一个语言模型，通常采用一元语言模型进行计算，查询语言模型与文档语言模型的距离可以用Kullback-Leiblerj距离来计算：

$$KL(Q,D) = \sum_{w_i \in Q} p_Q(w_i) \log \frac{p_Q(w_i)}{p_D(w_i)} \tag{8}$$

在语言模型的检索方法中涉及到一个重要的问题就是平滑算法，对语言模型进行平滑主要出于两个目的，首先如果某个查询词没有在文档中出现，那么这个查询词的概率 $p(w \mid d)$ 就为 0，这样在计算相似度时就会产生问题；其次语言模型的训练往往需要很大的训练样本，而一个文档的规模相对来说是非常小的，因此在采用最大似然估计时，会使某些概率估计不准确。这两方面的问题都可以通过语言模型的平滑技术得到比较好的缓解。针对信息检索问题，Simplified Jelinek-Mercer平滑方法、Dirichlet prior平滑方法和Absolute discounting平滑方法[8]以其较快的计算速度和较好的效果得到了比较多的应用。语言模型平滑的实质就是将在文档中能够被观察到的词的概率减掉一小部分，再将这部分概率重新分配给那些没有观察到的词。在Simplified Jelinek-Mercer平滑方法中将关键词在文档和文档集合中出现的概率进行加权处理，作为关键词概率的极大似然估计：

$$p(w \mid d) = (1 - \lambda) p_{ml}(w \mid d) + \lambda p(w \mid C) \tag{9}$$

Dirichlet prior 平滑方法中对于每个关键词都加上一个虚拟计数 $\mu p(w \mid C)$，这样关键词概率的极大似然估计可以表示为

$$\begin{aligned} p(w \mid d) &= \frac{c(w,d) + \mu p(w \mid C)}{\mid d \mid + \mu} \\ &= \frac{\mid d \mid}{\mid d \mid + \mu} p_{ml}(w \mid d) + \frac{\mu}{\mid d \mid + \mu} p(w \mid C) \end{aligned} \tag{10}$$

Absolute discounting 平滑方法中将关键词出现的次数减去一个常数 δ 进行平滑，计算方法如下：

$$p(w \mid d) = \frac{\max(c(w,d) - \delta) + \delta \mid d \mid_u \ p(w \mid C)}{\mid d \mid} \tag{11}$$

3 种概率平滑的方法都有各自的特点，不能说哪种平滑方法的效果更优秀[8]，具体的与文档集合、查询集合有一定的关系。

除 INQUERY 方法和上面介绍的一些经典的检索方法外，作为集合选择的方法应用得较少。虽然集合选择问题可以转化为一个文档检索问题，但考虑到二者之间的差异，这些检索方法在集合选择中的有效性需要进一步的研究验证。

3 实验及分析

将上面介绍的各种文档检索方法分别应用在文档检索和集合选择问题上，对各种方法的实验结果进行对比。实验采用的数据集合是TREC中Web track使用的Gov数据集合，在分布式信息检索中Gov数据集被划分成 100 个子集合，查询集是Web track2002 的查询。对于tf.idf方法分别应用了原始tf值raw(tf)、对数tf值log(tf)、cosine规格化 3 种不同的形式，对于语言模型的方法分别采用了JM、Dir、Abs 3 种不同的数据平滑方法进行检索，实验采用的检索工具是经过定制的Lemur[9]检索系统。

在对检索结果进行评价时，文档检索采用了平均准确率

(下转第 210 页)