●陆 伟 [英]Stephen Robertson

基于域加权词频法的 XML 文档级检索实现与评价*

摘 要 利用 BM25F 模型,通过实验,在 INEX 04 数据集的基础上,实现了对多个域(元素)词频进行加权的 XML 文档级检索。XML 文档结构的确蕴含了一定的语义信息。利用这些语义信息,可以提高检索性能。表 2。图 1。参考文献 16。

关键词 可扩展标记语言 检索 域加权词频法 BM25F 模型 INEX 04

分类号 G354

ABSTRACT With a BM25F model and experiments based on INEX 04 data sets, the authors realize field-weighted XML document level retrieval. The XML file structure implies some semantic information, which can be used to improve the performance of retrieval. 1 tab. 1 fig. 16 refs.

KEY WORDS XML. Retrieval. Field-weighted term frequency method. BM25F model. INEX 04.

CLASS NUMBER G354

与传统的文本信息检索不同的是,XML 不仅仅要求文档级的检索,而且需要实现元素级的检索。然而,即使是文档级的检索,它也与普通本文信息检索有所不同。XML 文档通常包含一些子域(元素),如 IEEE 提供的 INEX 数据集[1]就包括 title, abs, bdy, bm, st 等,实践证明探讨文档的内部结构对提高检索性能是有帮助的。因而一些研究人员试图寻找一些方法用以利用文档的这些内部结构。笔者曾提到主要有 3 种计算文档权重的方法[2],其一是简单的用某一域的权重代替整个文档权重,其二是用各个域权重得分之和来作为文档权重,另外一种就是采用如 Robertson 等提出的域词频加权法来计算文档权重[3]。第一种方法比较简单,也存在很大问题,这里不做讨论。第二种方法目前被很多学者采用,如 Wilkinson 和 Ogilvie 等提出并检验了一些计算域权重并合并这些域的分值以计算文档权重的方法[4~5];Kraaij 等提出了基于语言模型的较灵活的算法,然而并未具体实现[6];Myaeng 等利用贝叶斯网络合并不同文档内的词[7]。Robertson 等对该领域的相关研究做了详细评述[8]。

然而由于关于第二种方法的一些排序比较复杂,许多系统在实际实现上都是采用线性合并每一个域的权重得分的方法。Robertson 等从词频、文档长度、域值合并等角度分析了这种方法的危险性并提出了第三种方法,即线性合并每一个域的加权词频,然后利用 BM25 计算给定文档权重(本文中把这种基于 BM25 的域词频加权算法称为 BM25F)。他们在两个数据集"Reuters vol. 1"和"2002 TREC Web-Track crawl of the . gov"的基础上对一至两个域(标题和链接文本)的词频进行加权的实验,显示该方法简单可行,容易理解,并且确实在实验中提高了检索性能。而对于复杂的 XML 文档如 INEX 的 IEEE 数据集等往往有多个有价值的域需要予以考虑,该模型的效果如何,尚有待具体实验的验证。本文将以 INEX 04 的 IEEE 数据集 INEX 1.4 及相应的 ad hoc 检索主题与各主题的相关结果集为基础,应用 BM25F 模型对多个域(3 个)词频进行加权,实现 XML 的文档级检索并对检索效果进行验证。要实现这一目标,难度有两个:其一是域的选取,其二是对各个域权重的确定。与文献[9]一样,本文同样采用的是调适的方法,但由于选择域的增加,需要对每个域的权重值进行调适,增大了系统运算的难度和复杂度。

1 BM25F 模型

1.1 BM25 模型

BM25F 模型是 BM25 的域加权版本,是由 Robertson 等在 BM25 模型的基础上提出的[10]。BM25 模型是典型的概率检索模型,它包含多个变种模型,用以实现不同的信息检索目的,如 BM25b,BM250,BM251 等,也是知名的检索实验系统 Okapi 检索模块的核心模型[11]。对于 ad hoc 检索来说,如果忽略查询语句中查询词重复的现象,BM25 模型可以简化为[12]:

$$w_j(\vec{d},C)=\frac{(k_1+1)tf_j}{k_1\left((1-b)+b\frac{dl}{avdl}\right)+tf_j}\log\frac{N-df_j+0.5}{df_j+0.5} \qquad (1)$$

其中 C 指文档集,tf_j 是查询语句中第 j 个词在文档 $\vec{d}$ 中的词频,df_j 是词 j 的文档词频,即该词在 C 中出现的文档数目,dl 是文档长度,$avdl$ 是文档集中平均文档长度,k_1 和 b 在该模

* 本文为国家社科基金项目(编号 04BTQ016)和湖北省科技攻关项目(编号 2004AA101C99)成果。

型中是可变参数,其值可根据不同的文档集进行调整。

据此可以得出给定查询语句 q,文档权重得分为:

$$W(\bar{d},q,c) = \sum_j w_j(\bar{d},C) \cdot q_j \qquad (2)$$

1.2 BM25F 模型

BM25 并没有考虑文档的结构信息,对于一个查询词,不管出现在文档的哪一部分,重要性都是一样的。而对于 XML 文档来说,不同的结构往往蕴涵了不同的语义信息,因而给定查询词,在不同结构中出现的重要性有时并不相同,如该词出现在文章标题和正文中的重要性有明显区别。这就需要考虑查询词在不同域中的权重问题,Robertson 等据此提出了域词频加权的权重计算公式 BM25F[13]:

$$wf_j(\bar{d},C) = \frac{(k'_1+1)tf'_j}{k'_1\left((1-b)+b\frac{dl'}{avdl'}\right)+tf'_j}\log\frac{N-df_j+0.5}{df_j-0.5} \qquad (3)$$

其中 tf'_j 指加权后的第 j 个查询词在文档 $\bar{d}$ 中的词频,dl' 是加权后的文档长度,$avdl'$ 是加权后的平均文档长度,k'_1 是加权后的自由参数。

假设在给定文档 d 中有 nF 个域 $f=1,\cdots,nF$,查询词 j 在域 f 中的词频为 $tf_{d,j,f}$,则有多种途径去定义一个域的长度,最简单的就是用该域内所包含的可索引词的个数,即

$$dl_f = \sum_{j\in V} tf_{d,j,f}$$

其中 V 代表该文档集内所有可索引词的集合。

如果不考虑域加权的情况,查询词 j 在整个文档内的词频为

$$tf_{d,j} = \sum_f tf_{d,j,f}$$

而文档长度则为

$$dl = \sum_f dl_f = \sum_f \sum_j tf_{d,j,f} = \sum_j tf_{d,j}$$

相应的平均文档长度为

$$avdl = \frac{1}{N}\sum dl$$

在考虑到域加权的情况下,假设域词频为 W_f,则相应的参数变为

$$tf'_{d,j} = \sum_f W_f tf_{d,j,f}$$

$$dl' = \sum_f W_f dl_f = \sum_f \sum_j W_f tf_{d,j,f} = \sum_j tf'_{d,j}$$

$$avdl' = \frac{1}{N}\sum dl'$$

$$k'_1 = k_1\frac{atf_{weighted}}{atf_{unweighted}} = k_1\frac{avdl'}{avdl}$$

对于各域权重相加的算法,以 BM25 为例,思路是将 BM25 模型公式直接用于计算各域的权重得分(当然词频为域词频),然后将各个域的权重得分以某种形式相加,一般是线性相加,获取整个文档的权重得分。Robertson 等已经对此做了详细评述[14]。

2 实验数据集的选择及检索结果评价方法

2.1 实验数据集

(1) INEX 1.4 是 INEX 2004 年 ad hoc 检索的正式数据集。它由美国电气和电子工程师学会计算机学会提供,数据集中包含从 1995 年到 2002 年的各类文章共 12107 篇。它包含 XML 元素 8239873 个,属性 2204688 个,平均文档深度为 8。整个数据集为 XML 格式,共 494MB。

作为一个学术性的数据集,其中大部分文章都包含一些代表文章标题、摘要、正文、章节、章节标题、段落、参考文献及附录等的域(元素)(如表 1)。

表 1 INEX 1.4 数据集的重要域(元素)及含义

代表内容	域名(元素)
文章标题	atl
摘要	abs
正文	bdy
章节	sec,ssl,ss2,ss3
章节标题	st
段落	ilrj,ip1,ip2,ip3,ip4,ip5,item-none,p,p1,p2,p3
参考文献	bib
附录	bm

虽然在理论上应该对 XML 文档中所有域的权重进行调适,但在具体实现上,这将非常困难,对于像 INEX 1.4 这样的文档集来说,将花费难以接受的大量时间,而且这样做也没有必要。问题的关键就落在如何选择合适的域以对其权重进行调适。考虑到 INEX 1.4 文档集的结构特点,我们选择了文章标题 atl、摘要 abs 和章节标题 st 这 3 个域作为调适的对象。这 3 个域相对比较重要,我们有理由相信文章标题和摘要在一定程度上反映了该文章的内容,而章节标题在一定程度上反映了所在章节的内容。对于文中其他所有元素,都视为同等重要,其域权重都为 1,即不调适。

(2) INEX 04 ad hoc 检索主题:2004 年 ad hoc 检索主题共有 74 个,分为 CO(Content Only)和 CAS(Content and Structure)两类,其中 40 个为 CO 主题,34 个为 CAS 主题。CO 和 CAS 主题的区别在于后者对检索主题进行了结构限制。下面给出了这两类检索主题的差异:

<title>content based music retrieval</title>//CO 主题,关于基于内容的音乐检索。

<title>//article[about(.//atl, new book review bookshelf)]//sec[about(., database "data warehouse")]</title>//CAS 主题,要求文章标题关于新书评价,而章节是关于数据库和数据仓库的元素。

本文由于只探讨文档级的 XML 检索,因而只采用了 CO 主题。对于每一个检索主题,一般都包括主题编号、类型、主题(title)、描述及关键词等信息。同正式的 INEX 要求一样,我们只利用主题 <title> 项来作为检索输入。

(3)INEX 04 相关结果集:对于每一个检索主题,都有一个相关结果集,与检索主题一样,该结果集也是由每一个 INEX 04 参加者人工筛选评价而来的。相关结果集既用来评价每个参加者检索模型的效果,也可以作为进一步的研究之用。

INEX 相关结果集是以 XML 格式存储的,每一个主题都有一个相应的 XML 文件用以存储其相关结果集。由于 INEX 要求的是元素级检索,因而其结果集中存储的都是相关元素信息,如下给出了 2004 年第 162 号检索主题的相关结果集格式[15]:

```
<assessments pool = '58' topic = '162'>
  <file file = 'an/1995/a4086'>
    <path path = 'article[1]/bdy[1]/sec[1]' exhaustiveness = '0' specificity = '0' inpool = 'true' />
    <path path = '/article[1]/sec[1]/p[1]' exhaustiveness = '0' specificity = '0' inferred = 'true' />
    ...
  </file>
  <file file = 'an/1996/a1067'> ... </file>
  ...
</assessments>
```

这与文档级的检索评价要求不同。本文的实验需要一个文档级的相关结果集。因此我们开发了一个解析程序,利用该相关结果集产生本文实验所需的结果集。因为可以认为,文档中的元素相关则文档相关,只需提取出相关结果集中的文档信息即可。

2.2 检索结果评价方法

检索结果评价的两个重要指标是查全率和查准率。然而在实际评价的过程中,针对不同的数据集、不同的检索目标和任务,专家们提出了多种检索结果评价的方法,如 Fmeasure、Average Precision 以及 RPrecision 等。本文采用 Average Precision 评价方法[16],其公式为:

$$AveP = \frac{\sum_{j \in n} \frac{R_j}{j} C_j}{N} \qquad (4)$$

其中 n 是检索所返回的记录数,N 代表文档集中相关文档的个数,R_j 指第 j 个记录前返回的相关文档个数,C_j 是一个二元值,如果第 j 个记录相关,该值为 1,否则为 0。从公式可以看出,Average Precision 方法是建立在查准率基础上,又在一定程度上考虑了查全率问题的评价模型。本实验中,应用该评价方法计算所有检索主题的 *AveP*,然后取所有查询主题 *AveP* 的平均值,以判断设定相应域权值情况下的结果优劣。

3 实验过程及结果评价

本实验是在 Okapi 系统的基础上,在 Linux Red Hat 9.0 环境下进行的。详细步骤如下。

(1)文档预处理。在实验前,首先需要对 INEX 04 文档数据集进行归并,生成一个 Okapi 支持的格式. bib 文件。同时需要编制一个程序,将该文件内的每条记录转换成相应的 5 个域:atl, abs, st, bdy 和 bm,以支持域加权检索。这里一个需要特殊处理的是将所有章节的标题都归并到 st 域中。

(2)生成索引。利用在一定程度上支持 XML 索引的 Okapi 2.5.2 对该数据集进行索引,生成相应的倒排文档文件。

(3)检索结果。笔者开发了一个查询主题解析程序,对 INEX 04 的 CO 主题进行解析,然后利用Okapi的检索模块,获取初步的结果记录集,并存储在相应的临时文件中。(由于只是对模型进行检验,实验中并未考虑检索过程的效率问题。)

(4)结果排序。根据 BM25F 模型的思想,开发了一个相应的文档权重计算接口系统,用以对检索结果进行排序。该系统允许设定各域(atl, abs 和 st)的词频权重 W_f。

(5)结果评价。采用 Average Precision 评价方法,利用 INEX 相关结果集对排序结果进行评价,获取给定词频权重 W_f|atl, abs, st| 情况下的 *AveP* 值。由于有 40 个查询主题,因而最后的 *AveP* 为各个查询主题 *AveP* 平均值。为了便于对各域词频权重进行调适,笔者开发了一个调用软件可以自动设定各域的 W_f 区间,以获取最优 W_f 值。

需要说明的是,实验所采用的 W_f 值全部是正整数。在实验中,首先在|1,1,1| 到|10,10,10| 间调适 W_f|atl, abs, st|,每次对每个域的增量值为 1。实验结果显示,W_f|10,3,10| 的 *AveP* 值最大,此时 atl 和 st 域的权值都为最大值 10,而当 abs 域在权值为 3 ~ 6 之间时,*AveP* 值较大。因而实验进一步扩大了 atl 和 abs 的调适范围,进而设定 W_f|atl, abs, st| 的调适范围为|1,1,1| ~ |50,10,50|,同样增量为 1。这次的结果显示 st 的权值最优范围在 12 和 25 之间,而 atl 的值在 50 及其附近时最优,abs 的最优范围基本上没有变化。笔者决定将 abs 和 st 的 W_f 值分别设定在 1 ~ 10 和 10 ~ 30 之间,并进一步扩大范围调适 atl,在实验中使 atl 的增量为 10 在 1 ~ 300 之间调适并进一步设其增量为 50 一直调适到 3000。此时,结果显示无需再进一步调适下去,当 atl 在 2400 左右时,检索结果达到了峰值,如图 1 所示。该图显示了当 W_f(abs) 值为 4 时,分别在 0 ~ 3000 和 11 ~ 25 之间调适 W_f(atl) 和 W_f(st) 时 *AveP* 的变化。

最后的实验是在 2100 ~ 2700 之间以增量 1 的速度调适 atl,结果显示 W_f|atl, abs, st| 的值为|2356,4,22| 时 *AveP* 达到最大值,表 2 显示了实验中的部分调适结果。

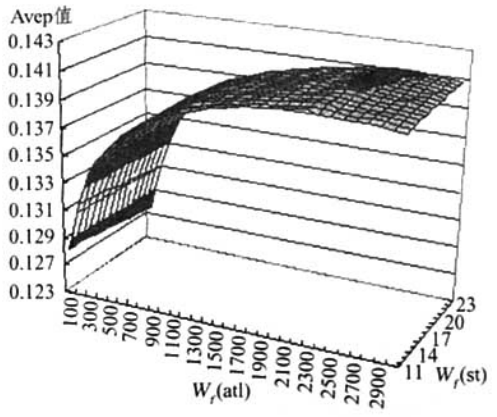

图1 *AveP* 值随 W_f(atl)和 W_f(st)的调适变化

表2 部分实验调适结果

W_f{atl, abs, st}	AveP	变化比率
2356,4,22	0.143698	+16%
1000,5,20	0.141485	+14%
100,3,13	0.135849	+10%
10,4,9	0.129819	+5%
1,1,1	0.124023	-

实验结果揭示了两点:其一,利用BM25F模型,对选定特定域词频加权是可行的,它在一定程度上提高了检索性能。从表1可以看到,当 W_f{atl, abs, st}的值为{1,1,1}时,即回到原来的BM25模型未加权的状态时,*AveP* 值为0.124023;而调适后的峰值出现在{2356, 4, 22}为0.143698,与前者相比,其检索性能提高了约16%。其二,XML文档结构的确蕴含了一定的语义信息,可以提高检索性能。对于本数据集来说,atl、abs和st的确如所预料的那样,对提高文档的检索性能有一定重要性。

实验结果也令人有些惊讶。在实验之前笔者预料到atl的值应该最高,但是没料到竟如此之高。一个可能的原因是本数据集来自学术性刊物,标题都比较规范,尽可能表达了文章的内容。但对此尚不能加以肯定,是否与检索主题的特殊性有关也还有待检验,这需要在更多检索主题的情况下做进一步实验。

4 结论

利用BM25F模型,在INEX 04数据集的基础上,本文实现了对多个域(元素)词频进行加权的XML文档级检索。实验结果证明,这一方法是可行的,在一定程度上提高了检索性能。然而,XML检索不仅仅是文档级的检索,用户需要更精确的相关元素信息。在进一步的研究工作中,如何以该模型的思想为基础,在对它适当改造的基础上,探讨它在元素级检索中的应用将是面临的又一个课题。截至本论文完成为止,我们已经在参加INEX 05的基础上对此做了一些工作,并提出了相应的元素级检索模型BM25E。然而,如何对模型中的参数进行修正和调适将是一个非常有挑战性的研究课题。笔者希望能够在此基础上进一步探索。

致谢

感谢国家留学基金委资助本文第一作者访学从事XML检索的相关研究工作,也感谢伦敦城市大学Andrew Macfarlane博士的意见和建议。

参考文献

1,15 INEX web site. [2006-03-29]. http://inex.is.informatik.uni-duisburg.de/

2 陆伟,夏立新.基于okapi的XML信息检索实现研究.中国图书馆学报,2006(4)

3,8,9,13,14 S. E. Robertson, H. Zaragoza, M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. CIKM'04, 2004

4 R. Wilkinson. Effective retrieval of structured documents. In Research and Development in Information Retrieval, 1994

5 P. Ogilvie, J. Callan. Combining document representations for known item search. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), 2003

6 W. Kraaij, T. Westerveld, D. Hiemstra. The importance of prior probabilities for entry page search. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002

7 S. Myaeng, D. Jang, M. Kim, Z. Zhoo. A flexible model for retrieval of SGML documents. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998

10,12 S. E. Robertson, S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. 1994, 345~354

11 S. E. Robertson. Overview of The OKAPI Projects. Journal of Documentation, 1997, 53(1)

16 G. Salton. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, 1989

陆 伟 武汉大学信息资源研究中心博士,副教授。通信地址:武汉。邮编430072。

Stephen Robertson 微软研究员,伦敦城市大学访问教授,交互式系统研究中心主任。(来稿时间:2006-04-12)