第35卷 第21期 计 算 机 工 程 2009年11月
Vol.35 No.21 Computer Engineering November 2009
·软件技术与数据库· 文章编号：1000—3428(2009)21—0085—03 文献标识码：A 中图分类号：TP311.52

一种相似重复元数据记录检测方法

王常武，韩菁华，张付志
(燕山大学信息科学与工程学院，秦皇岛 066004)

摘 要：对联邦数字图书馆中重复元数据记录进行检测和管理，是保证元数据质量、提高联邦检索服务质量的关键。针对现有联邦数字图书馆中重复记录检测方法计算集中、准确度不高等缺点，提出一种快速高效的相似重复元数据记录检测方法，该方法基于改进的 N-Gram 方法，适合较大规模联邦数字图书馆。模拟实验结果表明，该方法能有效提高重复检测的性能，加快重复检测的速度。

关键词：元数据；重复记录检测；N-Gram 方法；相似度

Method for Approximately Duplicate Metadata Record Detection

WANG Chang-wu, HAN Jing-hua, ZHANG Fu-zhi
(College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)

【Abstract】 Metadata records duplicate detection and management of federated digital library are one of key issues to ensure metadata quality and improve federal retrieval services. Many duplicate record detection methods exist for conventional federated digital library, but they are computationally intensive and low accuracy and so on. This paper proposes an efficient duplication approach for a relatively large federated digital library based on improved N-Gram method. Simulation experimental results show that the method improve the performance of duplicate detection effectively, accelerate the rate of duplicate detection.

【Key words】 metadata; duplicate record detection; N-Gram method; similarity

1 概述

遵从OAI模型的联邦数字图书馆提供的联邦检索服务是以利用OAI模型采集异构自治的数据提供者(DP)提供的元数据为基础，因此，集成的元数据质量直接影响信息服务的质量。在有关元数据质量的各种问题中，多数据源集成造成的元数据重复是最关键的问题之一，管理收集的重复元数据记录也是建立高质量联邦检索服务关键所在。

目前在相似重复记录检测方面已经有了一些研究成果。文献[1]提出用向量空间模型来计算元数据记录的相似性；文献[2]提出用滑动窗口技术计算元数据记录的相似性；文献[3]提出用 N-Gram 方法来进行相似重复记录检测。

上述方法都是对集中存放在本地数据库的元数据记录进行重复检测[4]，势必会造成处理的数据量过大，易形成性能瓶颈、检测速度不够快等缺点；而且上述所有算法都把元数据记录各个标签属性值视为同等重要的文本，没有体现元数据记录的特殊性，不能在较高层次上准确检测重复的元数据记录。本文针对这些缺点提出一种新的适合较大规模元数据仓储的相似重复记录检测方法。

2 相似重复元数据记录检测模型

针对较大规模联邦数字图书馆的相似重复元数据记录检测，本文提出一种相似重复元数据记录检测模型，该模型利用 Topic Model 算法进对收集到的元数据记录行聚类操作[5]，在产生主题的基础上，再利用 In-house 分类系统进行分类，在此基础上分发元数据分类分布存储，然后对分类存储的元数据记录进行相似重复检测。

在进行相似重复记录检测时，由于各个元数据记录标签在相似性比较上的重要度不同，因此本方法对不同标签赋予不同权重，而且在标签内部词性也表现出不同的重要度，对标签内部不同的词性也赋予不同的权重，从而能更精确地比较记录相似性。对 N-Gram 方法进行改进，在相似性计算中加入权重信息来进行相似性度量，称为加权 N-Gram 方法。整个相似重复元数据记录检测的结构模型如图 1 所示。

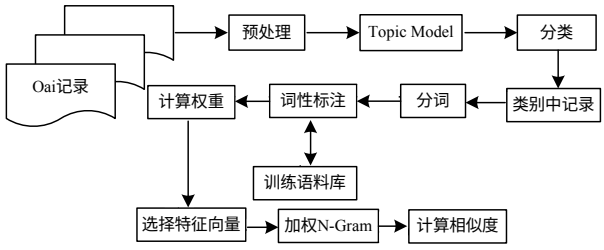

图 1 相似重复元数据记录检测结构模型

3 基于加权 N-Gram 相似重复元数据记录检测算法

基于加权 N-Gram 的相似重复元数据记录检测算法的基本思想是：提取元数据记录标签值及其属性值，并对标签值进行预处理，根据不同标签在重复检测中的重要度不同赋予不同标签不同的权重，称为标签权重；对标签内部根据词性计算词语权重，称为词性权重；综合词语各方面信息计算词语权重，依据权重信息选择特征向量，对特征向量进行 N-shingles 操作后，利用加权的 N-Gram 计算重复度。

基金项目：河北省自然科学基金资助项目(F2008000877)
作者简介：王常武(1970－)，男，副教授，主研方向：网格技术；韩菁华，硕士研究生；张付志，教授、博士生导师
收稿日期：2009-03-04 **E-mail：** xiaoshimei1980@163.com

3.1 权重计算

分别提取 Title, Creator, Subject, Description 标签的属性值，并分别按移除标点符号、删除停用词表中的词语等预处理规则对其预处理，而且相同标签的属性值连接，其他标签属性值连接为一个整体，为了描述称为 Connection 标签，并对不同标签赋予不同的权重，分别记为 $\pi(t)$，$\pi(a)$，$\pi(s)$，$\pi(d)$，$\pi(c)$，且 $\pi(t)+\pi(a)+\pi(s)+\pi(d)+\pi(c)=1$。其中，$0 \le \pi(c) \le \pi(d) \le \pi(s) \le \pi(a) < \pi(t) \le 1$。

Title, Creator, Subject, Description, Connection 标签的属性值中词性也能影响相似度计算，对它们内部的词性也赋予不同的权重，一般来说名词、动词、形容词比副词、代词、介词、连词更具有判别力。本算法采用增加了未登录词和收集了大量相似语料作为词性标注的训练语料库训练词性标注器，利用训练好的词性标注器对元数据记录中的词语自动标注词性。

确定一个词性权重区间[p, q]，p, q 是安排给同类词性权重的最小和最大值，且 $v \in G$，G 是具有相同词性的集合。则计算词性词语的权重公式可表示为

$$W(v) = \frac{(q-p)(F(v)-F')}{F-F'} + p \tag{1}$$

其中，$F(v)$是词语 v 在训练语料库中词频；F 和 F'分别是 G 中词性词语在训练语料库的最大和最小词频，而且长词语比短词语更重要；$|v|$表示词语 v 的长度；词语在元数据记录中的频率也不能忽略，$|f|$表示词频。综合以上考虑，则权重公式变为

$$W(v) = \frac{(q-p)(F(v)-F')}{F-F'}|v||f| + p|v||f| = \left(\frac{(q-p)(F(v)-F')}{F-F'} + p\right)|v||f| \tag{2}$$

根据上述方法，分别计算 Title, Creator, Subject, Description, Connection 中词性权重，并分别乘以权重系数，结果用 $W(t)$, $W(a)$, $W(s)$, $W(d)$, $W(c)$表示。

本算法根据权重选取特征向量，为使在重复记录检测中占据比较重要作用的 Title 和 Creator 标签中的属性值都能被选为特征向量，本文提出对 Title 和 Creator 标签中的权重再赋予一个增强因子 m, n，且 m 是满足式(3)的最小正整数；n 是满足式(4)的最小正整数。

$$m \times \min\{W(t)\} > \max\{W(s),\ W(d),\ W(c)\} \tag{3}$$

$$n \times \min\{W(a)\} > \max\{W(s),\ W(d),\ W(c)\} \tag{4}$$

则 Title 和 Creator 标签的权重变为

$$W(t') = m \times W(t) \tag{5}$$

$$W(a') = n \times W(a) \tag{6}$$

根据上述方法，计算各个标签词语权重，具有比较高权重的词语选为记录的特征向量，合并特征作为整个记录的特征向量，并按权值降序排列。

3.2 相似度计算

预先根据经验定义一个重复度阈值为 θ，当 2 个记录的相似度大于等于 θ 时，认为它们是相似重复记录，在文献[3]中，相似度用 Jaccard(用 4-shingles)相似性来定义，而本文为能充分利用元数据记录提供的各项信息，采用加权的 N-Gram 方法对元数据记录进行相似度计算。式(7)为记录 A 和记录 B 的相似度计算公式如下：

$$Sim(A,B) = \frac{\sum_{v_i \in R(A) \cup R(B)} W(v_i)}{\sum_{v_j \in R(A) \cup R(B)} W(v_j)} \tag{7}$$

判定 2 条记录是否是相似重复记录的算法描述如下：

输入 读入分类的元数据库，重复度阈值 θ

输出 true/false

```
1. pDict->Load("meta/dictionary.pdat"); //导入分词词典
2. PAlign->Load("meta/dictionaryAlign.aln"); //导入对齐字典
3. fopen("meta/wordweight.dat","rb"); //导入单词权重
4. fopen("meta/wordFrequency.dat","rb"); //导入频次文件
5. for(i=0;i<count-1;i++)
6. for(j=i+1;j<count;j++) //count 表示记录数
7. getString('metadata'); //提取 mysql 数据库中 metadata 字段值
8. pRdr->putContentHandler() //解析 metadata 字段值，并解析出
//各个标签的属性值
9. Sim(ri, rj); //计算 2 个记录的相似性
10. if Sim(ri,rj)≥θ
11. return true;
12. else
13. return false;
```

其中，计算 2 条记录的相似性 *Sim* 的算法如下：

输入 记录 r_1 和记录 r_2 的特征向量

输出 相似度值

```
1. if (*pos1==*pos2) //计算 2 个特征向量中相同值的权值
2. intersect+= r_fWordWeight[*pos1];
3. weight1+=r_fWordWeight[*pos1];
4. weight2+=r_fWordWeight[*pos1];
5. pos1++;
6. pos2++;
7. else
8. if (*pos1>*pos2)
9. weight1+=r_fWordWeight[*pos1];
10. pos1++;
11. else
12. weight2+=r_fWordWeight[*pos2];
13. pos2++;
14. if (weight1>weight2)
15. bigweight=weight1;
16. else
17. bigweight=weight2;
18. if (bigweight<0. 1)
19. return 0;
20. similarity=intersect/bigweight; //计算 2 条记录的相似度
21. return similarity;
```

上述算法中第 1 行~第 17 行是计算 2 个记录特征向量中交集权重和 *intersect* 以及并集权重和 *bigweight*。

3.3 相似对比优化

在进行两两比较之前，对 N 个临近特征进行 N-Gram 处理后，被 MD5 哈希成一个整数作为一个 Fingerprint，只有当 2 个记录有相同的 Fingerprint 时才进行比较，这样避免了没必要的对比操作，提高了计算速度。

为减少不必要的对比操作，假设相似重复关系具有弱传递性，即记录 A 和记录 B 相似重复，记录 B 和记录 C 相似重复，当 $Sim(A,B) \times Sim(B,C) > k$ 时，则 A 和 C 有关联性。其中，k 为使 A 和 C 具有关联性的阈值。

利用 Union-Find 结构计算相似传递闭包，把数据库中的

每条记录都视为无向图的一个顶点，每个顶点有如下结构：

Struct Node {long rid , int parent}

其中，*rid* 是记录标识；当 *parent*≥0 时，*parent* 表示父节点位置；当 *parent*<0 时，*parent* 表示根节点。

当 2 条记录被判定为重复记录时，将对应顶点所在的连通分量合并，任何时刻无向图中的连通分量就代表到目前为止检测到的相似重复记录的传递闭包。

4 模拟实验

为验证本文提出的检测重复元数据记录新方法的性能，从基地址为 http://arxiv.org/oai2(记录数 475 428 条)、http://www.citebase.org/oai(记录数 650 257 条)2 个数据提供者采集数据，搭建试验原型系统，本实验仅对 Title, Creator, Subject, Description 标签的属性值进行处理。用 2 个类别进行实验测试：Computer science 和 Society，其中，属于 Computer 的记录 arxiv 中有 12 419 条，citebase 中有 13 832 个；属于 Society 类的 arxiv 中有 2 230 条，citebase 中有 2 729 条。N-Gram 采用 3-shingles。

分布重复记录的检测时间以最迟完成检测的节点为准，图 2 为重复记录检测时间比较。可见，采用分布方式处理重复元数据具有更高的检测速度，而且这种处理方法更适合大规模的联邦数字图书馆元数据重复记录检测。

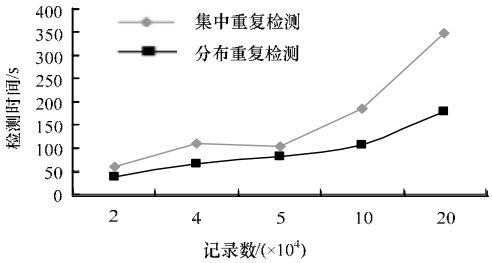

图 2 重复记录检测时间比较

图 3 为 Computer 类中标签权重对重复检测数目的影响。其中，横坐标轴 *a*, *b*, *c*, *d*, *e*,*f* 分别代表赋予标签不同权重系数的取值，如表 1 所示。可见，权重在 *d* 值范围时重复记录检测数达到最大值。

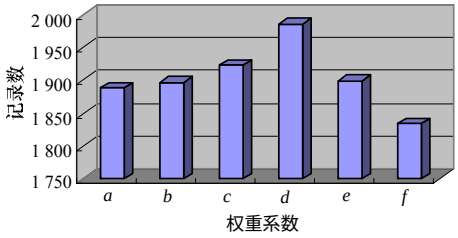

图 3 Computer 类中标签权重系数对记录数的影响

表 1 标签权重系数取值

标签	*a*	*b*	*c*	*d*	*e*	*f*
Title	0.2	0.3	0.4	0.5	0.6	0.7
Creator	0.4	0.2	0.3	0.3	0.2	0.1
Description	0.2	0.3	0.2	0.1	0.1	0.1
Subject	0.2	0.2	0.1	0.1	0.1	0.1

算法衡量指标用 *F*-measures 整体衡量：

$$F=\frac{2\times(B/C\times B/A)}{(B/C+B/A)}$$

其中，*C* 为分类元数据库中重复记录总数；*A* 为算法计算出的重复记录总数；*B* 为算法正确查出的重复记录数。召回率=*B*/*C*；准确率=*B*/*A*。

Computer 类和 Society 类的类重复检测结果如图 4 和图 5 所示。可见，当重复度=0.85 时，Computer 类和 Society 类的 *F* 值比较合适。

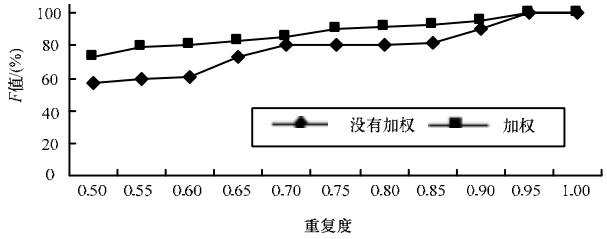

图 4 Computer 类重复检测结果

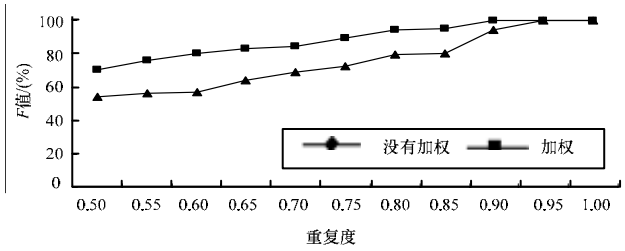

图 5 Society 类重复检测结果

5 结束语

为了解决现有比较大型联邦数字图书馆相似重复元数据记录检测计算集中，不适合处理大量记录的问题，本文对元数据重复检测采用分布并行检测的方法，提高了检测速度，并对现有重复检测算法进行改进，充分考虑元数据记录的结构特征，能快速准确地检测重复记录，有效提高了检测性能和速度。

参考文献

[1] Harrison T L, Elango A, Bollen J, et al. Initial Experiences Re-exporting Duplicate and Similarity Computations with an OAI-PMH Aggregator[R]. Norfolk, VA, USA: Old Dominion University, Tech. Rep.: cs.DL/0401001, 2004.

[2] Khan H M, Maly K, Zubair M. Similarity and Duplicate Detection System for an OAI Compliant Federated Digital Library[C]//Proc. of ECDL'05. Vienna, Austria: [s. n.], 2005.

[3] Foulonneau M. Information Redundancy Across Metadata Collections[J]. Information Processing and Management, 2007, 43(3): 740-751.

[4] Yang Hui, Callan J. Near-duplicate Detection by Instance-level Constrained Clustering[C]//Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA: ACM Press, 2006.

[5] Newman D, Hagedorn K, Smyth C C P. Subject Metadata Enrichment Using Statistical Topic Models[C]//Proc. of JCDL'07. Vancouver, Canada: ACM Press, 2007.

编辑 金胡考