doi:10.3969/j.issn.1673-9833.2014.06.016

基于分组指纹的细粒度相似性检测系统

盛鑫海[1]，袁鑫攀[1]，满君丰[1]，涂　慧[2]

（1. 湖南工业大学 计算机与通信学院，湖南 株洲 412007；2. 中南大学 信息科学与工程学院，湖南 长沙 410083）

摘　要：在文档相似性检测中，粗粒度会降低准确度，粒度过细又会大幅增加计算时间。针对基金项目相似性检测，在b位Minwise Hash算法的基础上，提出了一种细粒度文档相似性快速检测方法。先对文档进行预处理，提取文档正文，并生成分组指纹特征，再构建细粒度的分组指纹索引结构，利用海明距离来计算文档之间的相似性，以XML文档形式存储并显示相似信息。通过系统的实现，验证了该方法的有效性，且检索效率有所提高。

关键词：分组指纹；细粒度；文档相似性检测；海明距离

中图分类号：TP391.3　　**文献标志码**：A　　**文章编号**：1673-9833(2014)06-0081-05

The Fine-Grained Similarity Detection System Based on Grouping Fingerprint

Sheng Xinhai[1]，Yuan Xinpan[1]，Man Junfeng[1]，Tu Hui[2]

（1. School of Computer and Communication，Hunan University of Technology，Zhuzhou Hunan 412007，China；
2. School of Information Science and Engineering，Central South University，Changsha 410083，China）

Abstract：In document similarity detection, coarse grain will reduce the accuracy and too small particle size will increase the computation time. Proposes a quick document similarity detection method based on *b*-bit Minwise Hash. Firstly extracts the document text to generate a grouping fingerprint features; Then establishes the index structure of fine-grained grouping fingerprint; Finally computes the resemblance of document part by Hamming distance, and stores and displays the evidence of similarity by XML document format. Through system practice, verifies the effectiveness of the method and increases the efficiency of retrieval.

Keywords：grouping fingerprint; fine-grained; document similarity detection; Hamming distance

0　引言

随着信息时代的发展，数字文档（如基金项目申报文档、论文文档、网页等）呈几何级增长。由于数字文档具有易复制性，这导致项目重复申请、论文抄袭、网页重复等不良现象频频出现。大量相似文档的存在也降低了信息检索效率。因此，文档之间相似性的精确检测显得尤为重要。

在文档相似性检测中，经常出现一个文档的内容与另外一个或者多个文档具有一定的相似程度。其中，不仅仅只有两个文档完全一致的情况，还存在一些其他的相似，例如对其它文档的移位变换、重新组织文档结构等。假设一个文档F的内容全部

收稿日期：2014-09-25

基金项目：国家自然科学基金资助项目(61350011, 61402165)，湖南省自然科学面上基金资助项目（14JJ2115, 2015JJ3058），湖南省教育厅科技研究基金资助项目（14C0325），湖南工业大学自然科学研究基金资助项目（2014HZX17）

作者简介：盛鑫海（1987-），男，湖南岳阳人，湖南工业大学硕士生，主要研究方向为数据挖掘，E-mail：2213499@163.com

通信作者：袁鑫攀（1982-），男，湖南株洲人，湖南工业大学教师，博士，主要研究方向为信息检索和数据挖掘，E-mail：xpyuanfly@163.com

抄袭于文档集 $S=\{S_1, S_2, \cdots, S_n\}$，即 F 中的段落分别来源于文档集 S 中的文档，相当于文档 F 由集合 S 中的 $S_1, S_2, \cdots, S_n$ 拼凑而成。这种由多篇文档拼凑抄袭的现象是常见的。针对上述现象，若以文档级别作为相似性检测系统的检测粒度，则无法检测出文档 F 存在的段落或句子级的重复情况。因此，对于文档中的段落或者更加细粒度的句子的相似性检测具有十分重要的意义。

Minwise Hash[1]算法作为目前主流的海量集合相似度估计算法，在信息检索中得到广泛应用[2-3]。在Minwise Hash算法的基础上，学者们有了很多的理论创新。例如，b 位 Minwise Hash[4]算法降低了存储空间和计算时间，提高了算法效率；分数位 Minwise Hash 算法[5]对各种精度和存储空间需求有着更加广泛的选择性；连接位 Minwise Hash 算法[6]将位连接起来进行相似度度量，牺牲了算法 5% 的精度，却能成倍地减少比对的次数，提升算法的性能。Minwise Hash 算法还能应用于三者相似性检测[7-8]、大型线性支持向量机[9]以及基于最大似然估计改进的估计[10]算法。

本文以检测基金项目相似性为应用背景，以在海量数据环境中快速而准确地检测出文档的相似性为目的，提出了基于分组指纹的细粒度相似性检测系统。该系统在 b 位 Minwise Hash 算法的基础上，以“句子”细粒度来检测文档相似性。本文设计了检测系统框架，构建以细粒度为基本元素的分组指纹索引结构，并采用 XML 通用结构化的标记语言[11]作为相似性证据的存储结构。本文解决了系统的关键技术难点，为基金项目相似性检测系统的工程应用提供了理论依据。

1 系统架构

1.1 系统目标

本文的研究目标是针对各类自然科学基金项目申请书的内容特点，研究项目特征化与比对方法，开发能快速准确地发现抄袭、多头申请和多次申请的申请书相似度检测系统，解决基金项目形式审查中项目重复查证问题。以一份项目申报书作为查询条件，在基于分组指纹的细粒度相似性检测系统中进行相似度检测。该检测系统先从待检测项目申报书中抽取细粒度的指纹特征，然后在海量项目指纹特征库中进行检索，找到满足相似度阈值条件的项目文档。

文档相似性检测模型如图 1 所示。该模型包括：1）建立海量句子的分组指纹索引；2）提取待检测申请书的特征指纹，形成细粒度的指纹特征，利用分组检索算法检索文档相似性；3）对相似性项目的重复证据能够记录并呈现。

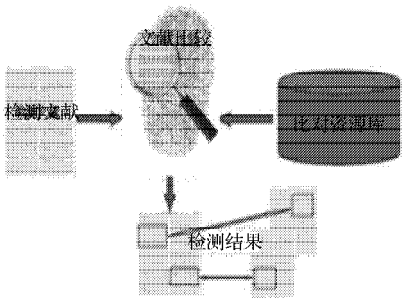

图1 文档相似性检测模型

Fig. 1 Document similarity detection model

1.2 系统功能结构

系统功能模块可以划分为文档预处理、相似性计算与相似性证据显示 3 大模块，如图 2 所示。

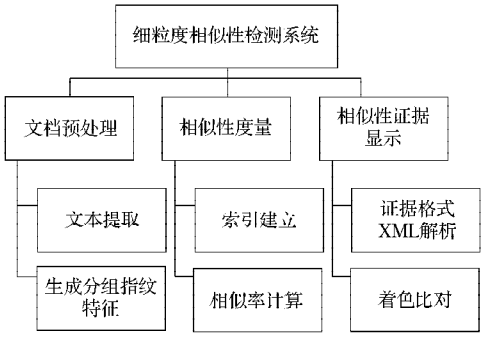

图2 功能结构图

Fig. 2 Functional block diagram

1）文档预处理

系统先提取待检测文档的正文（支持 Word 和 PDF 2 种格式），再生成分组指纹特征。生成分组指纹特征的具体步骤为：将提取的文档正文按照文章结构划分段落，将每一个段落划分成为“句子”的组合，对“句子”进行分词，提取 shingle，计算句子的 b 位 Minwise Hash 指纹。

2）相似度计算

索引建立。对每个“句子”的特征建立分组指纹索引。本文采用 Lucene 来建立索引，因为实验证明 Lucene 建立的索引，检索的速度更快。假设指纹特征存储在 Oracle 数据库，将 k=64 位指纹进行分组，分为 m=6 组（11,11,11,11,11,10）。分组数 m 势必比海明距离阈值 r_h 大，令 $c=m-r_h$。若每个分组命名为 $\{A_1, A_2, A_3, A_4, A_5, A_6\}$，分别对 6 个分组指纹建立 m=6 个 B+ 树索引，该算法的时间复杂度为 $O(\log(n))$。

相似率计算。每篇文档相当于一个“句子”集合。每个“句子”通过分组指纹索引，可以快速检索到相似的“句子”，采集相似证据并以相应的 XML 结构存储。

3）相似度证据显示

XML 解析。根据不同的证据显示模式，解析存储相似性证据的 XML 文档，再显示获取的相似性信息。

着色比对。通过 XML 的解析，将当前待查文档的相似信息高亮着色，同时对于两个相似性较高的文档进行着色比对。

1.3　系统拓扑结构

系统的拓扑结构如图 3 所示。用户上传待查文档，经过 Web Http 服务传输到检测引擎，建立“句子”细粒度待查索引，检测完毕后，显示文档相似性结果。

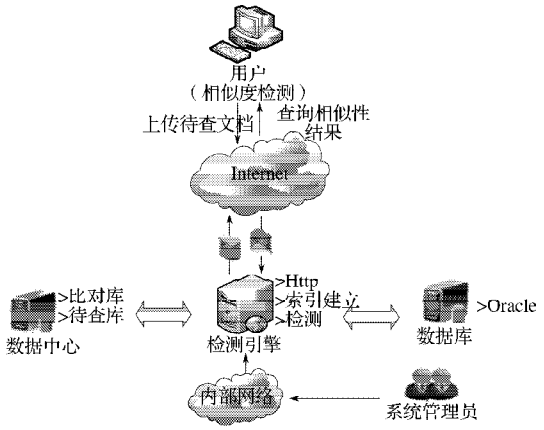

图 3　拓扑结构图

Fig. 3　The topological structure

1.4　数据处理流程

系统的数据处理流程如图 4 所示。后台程序首先对数据源进行数据处理操作，从特征计算开始，进行细粒度指纹的提取，通过 Lucene 工具[12]构建细粒度“句子”指纹索引，利用指纹分组快速检测相似度，并采集相似性证据，以 XML 格式结构进行存储；前台程序则解析 XML 文档，展示相似性证据。

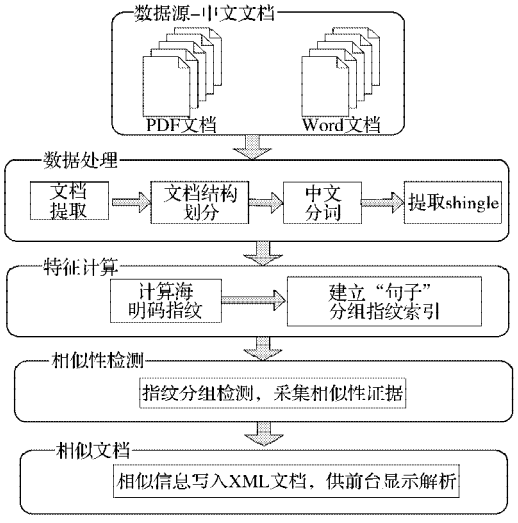

图 4　数据处理流程图

Fig. 4　The flow chart of data processing

2　系统关键技术

本文主要研究了文档的提取、特征化、低复杂性的指纹化、细粒度分组指纹检索算法、高效的比对技术和良好的相似性检索结果界面呈现等关键技术，如图 5 所示。

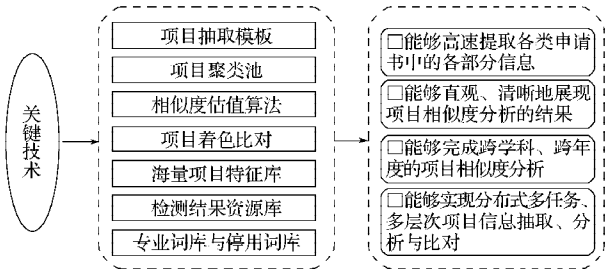

图 5　系统关键技术

Fig. 5　The key technology of system

2.1　确定检测粒度

细粒度文档相似性检测通常是先将文档切割为多个自定义长度的文本块集合，再通过相关检索，计算并获取每个文本块与文本集合中的文本的相似程度。如果文本块的长度选择过大，则计算准确度不高，容易遗漏多方抄袭部分内容的情况。如果文本块长度选择太小，就容易造成算法时间和算法空间的开销过大。

在文档切割的过程中，通常会按照自然段对文档进行初步划分，这是由于自然段可以表达作者相对完整的思想。另一方面，大部分抄袭者也都选择以段落或长句为单位进行抄袭。然而，文档中也常出现一些很长的自然段，完全基于自然段落的划分会降低检测精度。本文将自然段落切分为 150 字符左右的“句子”，将“句子”作为检测单元粒度。对于特殊的独句段和短句段等，不进行相似性的检测。因为这类段落通常具有较高的同义性，使用频度也很高，在文章中一般具有起承转合作用，与文档的实际内容无关。检测粒度划分流程如图 6 所示。

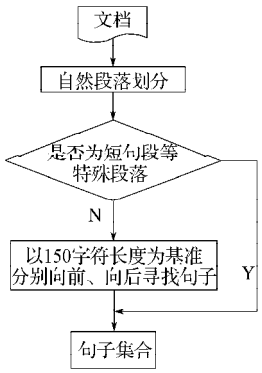

图 6　确定检测粒度流程图

Fig. 6　The flow chart of detection granularity

2.2 建立细粒度分组指纹索引

实际上，文档相似性检测就是检测与具有 s 个“句子”的文档 $\boldsymbol{D}'$ 相似度在海明距离范围内的文档集合，即检索文档相似度 R 大于阈值 r_{h} 的文档集合。故文档的相似性查询问题转换为：给定一个 k 位指纹的集合，查询该集合中与指纹 q 的海明距离在 r_{h} 以内的指纹集合。例如：当 k=100 时，检索文档相似度大于 80% 的文档集合，该集合为指纹海明距离在 r_{h} 以内的指纹集合。

令细粒度的检测单位“句子”的指纹为 k=100 位的海明码指纹（$f_{i,1}$~$f_{i,100}$），将 100 位指纹进行分组，分为 m=5 组{A, B, C, D, E}，各组的位数为（20, 20, 20, 20, 20, 20）[11]。具有 s 个长句的文档 $\boldsymbol{D}'$ 可表示为

$$\boldsymbol{D}'=\begin{bmatrix} f_{1,1},\cdots,f_{1,20} & f_{1,21},\cdots,f_{1,40} & \cdots & f_{1,81},\cdots,f_{1,100} \\ f_{2,1},\cdots,f_{2,20} & f_{2,21},\cdots,f_{2,40} & \cdots & f_{2,81},\cdots,f_{2,100} \\ \vdots & \vdots & & \vdots \\ f_{s,1},\cdots,f_{s,20} & f_{s,21},\cdots,f_{s,40} & \cdots & f_{s,81},\cdots,f_{s,100} \\ A & B & \cdots & E \end{bmatrix}。\quad（1）$$

文档数为 n 的文档集 M 表示为

$$\boldsymbol{M}=\begin{bmatrix} \begin{bmatrix} f_{1,1,1},\cdots,f_{1,1,20} & f_{1,1,21},\cdots,f_{1,1,40} & \cdots & f_{1,1,81},\cdots,f_{1,1,100} \\ f_{1,2,1},\cdots,f_{1,2,20} & f_{1,2,21},\cdots,f_{11,2,40} & \cdots & f_{1,2,81},\cdots,f_{1,2,100} \\ \vdots & \vdots & & \vdots \\ f_{1,s,1},\cdots,f_{1,s,20} & f_{1,s,21},\cdots,f_{1,1,s,40} & \cdots & f_{1,s,81},\cdots,f_{n,s,100} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} f_{n,1,1},\cdots,f_{n,1,20} & f_{n,1,21},\cdots,f_{n,1,40} & \cdots & f_{n,1,81},\cdots,f_{n,1,100} \\ f_{n,2,1},\cdots,f_{n,2,20} & f_{n,2,21},\cdots,f_{n,2,40} & \cdots & f_{n,2,81},\cdots,f_{n,2,100} \\ \vdots & \vdots & & \vdots \\ f_{n,s,1},\cdots,f_{n,s,20} & f_{n,s,21},\cdots,f_{n,1,s,40} & \cdots & f_{n,s,81},\cdots,f_{n,s,100} \end{bmatrix} \end{bmatrix},$$

式中 $f_{n,s,k}$ 为第 n 个文档的第 s 个句子对应的第 k 位海明码。

为 $\boldsymbol{M}$ 定义向量 $\boldsymbol{V}_A$, $\boldsymbol{V}_B$, $\boldsymbol{V}_C$, $\boldsymbol{V}_D$, $\boldsymbol{V}_E$，即

$$\boldsymbol{V}_A=\left[f_{1,1,1}\cdots f_{1,1,20},f_{1,2,1}\cdots f_{1,2,20},\cdots,f_{n,s,1}\cdots f_{n,s,20}\right],$$

$$\boldsymbol{V}_B=\left[f_{1,1,21}\cdots f_{1,1,40},f_{1,2,21}\cdots f_{1,2,40},\cdots,f_{n,s,21}\cdots f_{n,s,40}\right],$$

$$\vdots$$

$$\boldsymbol{V}_E=\left[f_{1,1,81}\cdots f_{1,1,100},f_{1,2,81}\cdots f_{1,2,100},\cdots,f_{n,s,81}\cdots f_{n,s,100}\right]。$$

文档集表示为 $\boldsymbol{M}=\left[\boldsymbol{V}_A^{\mathrm{T}}\ \boldsymbol{V}_B^{\mathrm{T}}\ \boldsymbol{V}_C^{\mathrm{T}}\ \boldsymbol{V}_D^{\mathrm{T}}\ \boldsymbol{V}_E^{\mathrm{T}}\right]$，分别对向量 $\boldsymbol{V}_A$, $\boldsymbol{V}_B$, $\boldsymbol{V}_C$, $\boldsymbol{V}_D$, $\boldsymbol{V}_E$ 建立 5 个 B+ 树索引。在实际的系统应用中，可以利用数据库管理技术在指定的表中建立 5 个字段，并对这 5 个字段分别建立 INDEX 索引。

2.3 检索相似性

将待查文档分为“句子”的集合，然后提取“句子”的指纹，将其作为检测的基本粒度单元，检索每个待查“句子”与指纹库的重复情况。设海明距离阈值 r_{h} 为 2，即 100 位中不得超过 2 位以上不同。检索文档相似性的具体步骤如下。

步骤 1 将待查文档的“句子”指纹分组为{A_q, B_q, C_q, D_q, E_q}。从 5 组中选出 2 组，不同组合共有 $\binom{m}{m-r_{\mathrm{h}}}=\binom{5}{3}=10$ 种。

步骤 2 分别在 5 个 B+ 树上执行查询条件式（2），执行查询条件后的候选集合为 Set（Pre_Query）。

$$\left.\begin{aligned} &\text{where}\left(A=A_q\ \text{and}\ B=B_q\right) \\ &\quad\text{or}\left(A=A_q\ \text{and}\ C=C_q\right) \\ &\quad\cdots \\ &\quad\text{or}\left(D=D_q\ \text{and}\ E=E_q\right) \end{aligned}\right\}=\binom{m}{m-r_{\mathrm{h}}}=10\text{种}，\quad（2）$$

查询条件（2）的时间复杂度是

$$O\left(\left(m-r_{\mathrm{h}}\right)\binom{m}{m-r_{\mathrm{h}}}\log n\right)。$$

由于候选集 Set（Pre_Query）中的数量远小于问题规模 n，从而可以避免大量的海明距离计算。

步骤 3 对 Set（Pre_Query）中的所有元素进行海明距离计算，将满足海明距离小于 r_{h} 的元素加入集合 Set($Query$) 中。

2.4 系统界面

相似性检索界面如图 7 所示。在检索条件区域，用户可以输入申报书的时间、单位等信息进行查询。分模块相似率查询区域中，可以对申报书进行多样化的检索，如申报书的项目基础信息相似度、整体相似度、章节粒度的相似度等。计算结果按照相似度由大到小排序。点击“详细”按钮，可以得到两个文档的详细比对结果。

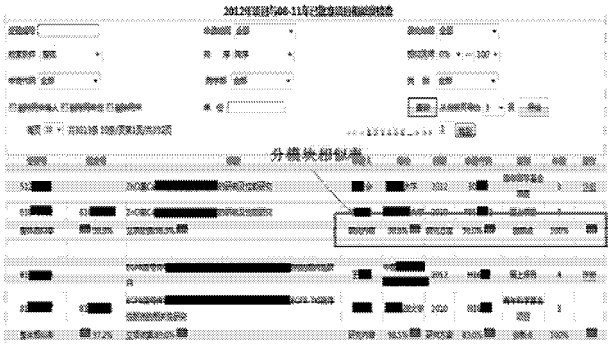

图 7 相似性检索界面

Fig. 7 The similarity retrieval interface

相似性证据界面如图 8 所示。其中包括一对一直观比对显示以及一对多的抄袭证据显示。

a）一对多

b）一对一

图8　相似性证据显示图

Fig. 8　Evidence of similarity

3　结语

Minwise Hash算法广泛应用于文本相似度检测。因此，针对基金项目相似性检测问题，在b位Minwise Hash算法的基础上，本文提出了一种细粒度文档相似性快速检测方法。该方法采用b位Minwise Hash算法进行相似度估值，生成特征指纹，并将相似度搜索问题转换为海明距离搜索问题，建立了分组指纹索引，灵活运用XML文档来保存相似证据。实验结果表明了该系统的有效性。

参考文献：

[1] Broder A Z，Charikar M，Frieze A M，et al. Min-Wise Independent Permutations[J]. Journal of Computer Systems and Sciences，2000，60(3)：630-659.

[2] Feigenblat G，Porat E，Shiftan A. Exponential Time Improvement for Min-Wise Based Algorithms[C]// SODA'11 Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms. San Francisco：SIAM，2011：57-66.

[3] Kane D M，Nelson J，Woodruff D P. An Optimal Algorithm for the Distinct Elements Problem[C]// PODS'10 Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. New York：ACM，2010：41-52.

[4] Li Ping，König A C. b-Bit Minwise Hashing[C]// Proceedings of the 19th International Conference on World Wide Web. [S. l.]：ACM，2010：671-680.

[5] 袁鑫攀，龙　军，张祖平，等. 最优分数位minwise哈希算法的研究[J]. 计算机科学，2012，39(8)：182-185.
Yuan Xinpan，Long Jun，Zhang Zuping，et al. Research on Optimal Fractional Bit Minwise Hashing[J]. Computer Science，2012，39(8)：182-185.

[6] 袁鑫攀，龙　军，张祖平，等. 连接位Minwise哈希算法的研究[J]. 计算机研究与发展，2013，50(4)：883-890.
Yuan Xinpan，Long Jun，Zhang Zuping，et al. Connected Bit Minwise Hashing[J]. Journal of Compuer Research and Development，2013，50(4)：883-890.

[7] Li Ping，König A C，Gui W. b-Bit Minwise Hashing for Estimating Three-Way Similarities[C]//Neural Information Processing Systems (NIPS). Vancouver：[s. n.]，2010：1387-1395.

[8] 袁鑫攀，盛鑫海，龙　军，等. 基于连接位Minwise哈希的三者相似性度量算法[J]. 上海交通大学学报，2014，48(7)：936-941.
Yuan Xinpan，Sheng Xinhai，Long Jun，et al. Connected Bit Minwise Hashing for Estimating Three-Way Similarities [J]. Journal of Shanghai Jiaotong University，2014，48(7)：936-941.

[9] Li Ping，Moore J，König A C. b-Bit Minwise Hashing for Large-Scale Linear SVM[J]. Neural Information Processing Systems，2011：101-109.

[10] Li Ping，König A C. Accurate Estimators for Improving Minwise Hashing and b-Bit Minwise Hashing[R/OL]. [2014-06-27]. http://arxiv.org/pdf/1108.0895.pdf.

[11] Bray T，Paoli J. Sperberg-McQueen：Extensible Markup Language(XML) 1.0[R/OL]. [2014-03-27]. http://www.w3.org/TR/REC-xml/.

[12] Yuan Xinpan，Long Jun，Zhang Zuping，et al. Near-Duplicate Document Detection with Improved Similarity Measurement[J]. Journal of Central South University，2012，19(8)：2231-2237.

（责任编辑：邓　彬）