自然语言文档复制检测研究综述*

鲍军鹏+, 沈钧毅, 刘晓东, 宋擒豹

(西安交通大学 计算机科学与技术系,陕西 西安 710049)

A Survey on Natural Language Text Copy Detection

BAO Jun-Peng+, SHEN Jun-Yi, LIU Xiao-Dong, SONG Qin-Bao

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

+ Corresponding author: Phn: 86-29-3042054, E-mail: baojp@mail.xjtu.edu.cn

http://www.xjtu.edu.cn

Received 2002-09-03; Accepted 2003-05-27

Bao JP, Shen JY, Liu XD, Song QB. A survey on natural language text copy detection. *Journal of Software*, 2003,14(10):1753~1760.

http://www.jos.org.cn/1000-9825/14/1753.htm

Abstract: Copy detection has very important application in both intellectual property protection and information retrieval. Currently, copy detection concentrates on document copy detection mainly. In early days, document copy detection concentrated on program plagiarism detection mainly and now the most studies are on text copy detection. In this paper, a comprehensive survey on natural language text copy detection is given, the developments of copy detection is introduced. The approaches and features of a variety of existing text copy detection systems or prototypes are reviewed in detail. Then some key detection techniques are listed and compared with each other. In the end, the future trend of text copy detection is discussed.

Key words: copy detection; plagiarism; intellectual property protection; information retrieval

摘 要: 复制检测技术在知识产权保护和信息检索中有着重要应用.到目前为止,复制检测技术主要集中在文档复制检测上.文档复制检测在初期主要检测程序复制,现在则主要为文本复制检测.分别介绍了程序复制检测和文本复制检测技术的发展,详细分析了目前已知各种文本复制检测系统的检测方法和技术特点,并比较了各系统关键技术的异同,最后指出了文本复制检测技术的发展思路.

关键词: 复制检测;剽窃;知识产权保护;信息检索

中图法分类号: TP309 **文献标识码**: A

复制检测(copy detection)又称剽窃检测(plagiarism detection),也有人称为副本检测(duplicate detection),不但是实施知识产权保护(intellectual property protection)的一种重要手段,也是提高信息检索(information retrieval)效率的一种手段.所谓复制检测,就是判断一个文件的内容是否抄袭、剽窃或者复制于另外一个或者多

* Supported by the National Natural Science Foundation of China under Grant No.60173058 (国家自然科学基金); the Science Research Foundation of Xi'an Jiaotong University of China under Grant No.573031 (西安交通大学科学研究基金)

第一作者简介: 鲍军鹏(1974－),男,陕西咸阳人,博士生,讲师,主要研究领域为人工智能,数据挖掘.

个文件.剽窃不仅仅意味着原封不动地照搬,还包括对原作的移位变换、同义词替换以及改变说法重述等方式.

目前数字知识产权主要有两种保护措施,一种是“阻止”法,另一种是“检测”法.“阻止”法就是使用加密、水印、特殊载体等方法,使受保护内容难以拷贝.例如,IEEE 通过光盘发行文集,中国期刊网上的文章采用专用软件才能阅读.文献[1]介绍的“安全打印机”使用加密的安全通信途径;文献[2]提出的“主动文档”需要使用专用程序;贝尔实验室提出了“水印”技术,使用加密的单词空格或者图像,可以鉴定文档授权用户身份[3,4].但是,上述方法都有可能被破解,而且我们也缺乏技术手段来防止授权用户使用光学识别(OCR)等办法去非法复制、扩散.所以,“阻止”法不能完全解决知识产权保护问题.

用“检测”法保护知识产权的思路是这样的:它并不关心文件是如何被复制的,而是首先判断出当前的文件中是否含有复制或者剽窃的内容,如果发现了非法复制或者剽窃行为,那么再对复制源或者剽窃者采取相关措施.“检测”法的核心就是复制检测技术.显然,“阻止”法和“检测”法不是相互对立的关系,而应该相互补充、完善才能更好地保护知识产权.

复制检测技术还可以提高信息检索效率.在信息检索中,我们总是希望尽快找到需要的内容,减少返回的无用内容.例如,一篇文献可能以 pdf,ps,word,html 等多种格式存在于多个不同网站上.当我们在 Internet 上搜索文献时,极有可能从多个地址返回内容相同的结果.这显然既浪费网络资源和检索资源,也浪费检索者的精力和时间.然而目前的各种信息检索方法和工具(例如 Google,Yahoo,Ei 以及 IEEE 等网站上的检索工具)都只是返回相关(符合查询条件)文章,而不能保证返回结果是否与已有内容重复.显然,在以后的信息检索中也应该加入复制检测技术.

1 复制检测技术分类

数字产品主要有文档、图像、音频、视频这 4 种表现形式.然而,复制检测基本上都集中在文档检测上.文档复制检测分为两类,一类是程序复制检测,另一类是自然语言文本复制检测.文献[5]报道了在 VLSI 设计中 CAD 文件上的复制检测,但其使用的方法与程序复制检测类似.程序复制检测和自然语言文本复制检测有很多方法是相似的,有的检测工具能够同时检测程序复制和文本复制.例如,悉尼大学 Wise 开发的 YAP3[6].但是,二者也有不同之处:所有的计算机程序都有严格的形式化语法,但是自然语言文本不受形式化语法限制,含义模糊的表达到处都是;程序中标识名称可以随意替换而语义不变,但是自然语言中的字词不能任意替换;程序的结构信息清晰,容易获取,而自然语言文本的结构特征一般不明显.总之,自然语言文本复制检测技术相对更难一些.

图像、音频、视频类文件的数据量远远大于文档类文件,其文件内容无法像字符串一样直接匹配、比较.现在关于这 3 类多媒体文件的基于内容的检索方法正处于积极的探索之中.显而易见,多媒体文件内容检测方法与文档文件内容检测方法相去甚远,本文不作介绍.其有关评述可参见文献[7~9].

2 复制检测技术的发展

2.1 程序复制检测技术的发展

最早在 20 世纪 70 年代初就有学者研究阻止大规模拷贝程序的技术和软件. Ottenstein 在 1976 年首次提出了基于属性计数法(attribute counting)检测软件剽窃的方法.但是,单纯的属性计数法抛弃了太多的程序结构信息,导致错误率太高.Verco 和 Wise[10]在 1996 年指出,对于仅仅使用属性计数法的检测算法,增加向量维数并不能改善错误率.改进属性计数法的措施就是加入程序的结构信息,结合结构度量(structure metrics,也称为控制流(control-flow))来检测剽窃.现在检测程序复制都是用各种方法综合属性计数和程序结构度量[11~13].Parker 等人[14]和 Clough[15]分别对上述的各种程序复制检测方法作了详细的介绍和评述.此外,还有人提出用神经网络来检测程序复制[16].

2.2 自然语言文本复制检测技术发展

自然语言文本复制检测技术的出现比程序复制检测晚了 20 年.1993 年,ARIZONA 大学的 Manber 提出了

一个 sif[17]工具,用于在大规模文件系统中寻找内容相似的文件.Sif 工具并未明确提出要进行文本复制检测.但是,sif 工具提出了“近似指纹(approximate fingerprints)”,就是用基于字符串匹配的方法来度量文件之间的相似性.这个思路被很多后来的文本复制检测系统所采用.1995 年,Stanford 大学的 Brin 和 Garcia-Molina 等人在“数字图书馆”工程中首次提出了文本复制检测机制 COPS(copy protection system)系统[18]与相应算法.COPS 系统框架为以后的自然语言文本复制检测系统奠定了基础,后来的检测系统框架与 COPS 大同小异. Garcia-Molina 和 Shivakumar 等人又提出了 SCAM(Stanford copy analysis method)原型[19,20]改进 COPS 系统,用于发现知识产权冲突.SCAM 借鉴了信息检索技术中的向量空间模型(vector space model)[21],使用基于词频统计的方法来度量文本相似性.后来 Garcia-Molina 和 Shivakumar 等人还在 SCAM 的基础上提出了 dSCAM 模型[22],把检测范围从单个注册数据库扩展到分布式数据库上以及在 Web 上探测文本复制的方法[23].同期,贝尔实验室的 Heintze 开发了 KOALA 系统[24]用于剽窃检测.KOALA 系统采用与 sif 基本相同的方法,与之类似的方法还有 Broder 等人提出的“shingling”方法[25].香港理工大学的 Si 和 Leong 等人建立的 CHECK 原型[26]采用统计关键词的方法来度量文本相似性.但是 CHECK 系统首次把文档结构信息引入到文本相似性度量中.到了 2000 年,Monostori 等人建立了 MDR(match detect reveal)原型[27~30].MDR 用后缀树(suffix tree)来搜寻字符串之间的最大子串.后来,Monostori 等人又提出用后缀向量(suffix vector)存储后缀树[31].西安交通大学宋擒豹等人提出了 CDSDG(copying detection system of digital goods)系统[32],这是为了解决数字商品非法复制和扩散问题而开发的一个基于注册的复制监测原型系统.悉尼大学 Wise 开发了 YAP(yet another plague)1,YAP2,YAP3 系列工具[6].YAP1 和 YAP2 是用于程序复制检测的工具,YAP3 利用程序复制检测的方法,既检测程序复制也检测文本复制.Glatt[33]检测剽窃的方法与众不同,需要被检测人参与测试.Glatt 认为每个人都有自己独特的写作风格,这个写作风格就可以作为“指纹”,并且每个人比其他人更清楚自己的写作风格.所以,Glatt 在 Wilson Taylor’s (1953) 完形填空程序的基础上建立一个程序,在一篇文档中去掉一些单词留出空白,然后叫被测试人补空.最后补空的正确率就是评估剽窃的依据.这个方法显然不够自动化,有些繁琐.

目前 Internet 上还有一些提供自然语言文本复制检测服务的网站和工具.例如,Plagiarism.org[34],EVE2[35]网站和 WordCheck[36]软件.另外,Jplag[13]和 MOSS(measure of software similarity)网站[37]提供程序复制检测服务.这些网站都没有详细介绍其具体的检测算法,提供的服务和功能也各有特色.关于这些服务和软件的评测参见文献[38~40].

3 自然语言文本复制检测中的几个问题

3.1 文本特征问题

3.1.1 特征提取方式

文本复制检测的核心内容就是判断两篇文本内容是否存在雷同成分,并给出一个数值评估.这个数值我们可以称为相似度.相似度越大,文本雷同成分越多;相似度越小,文本雷同成分越少.显而易见,个别语句相同的文本不能算作剽窃;只有当两篇文本的相似度大于某个阈值时,才能判定为剽窃.在计算文本相似度时,首先需要提取文本的特征.

根据提取文本特征的方式,我们把文本复制检测方法分为两类.一类采用基于字符串比较的方法,也称为基于语法(syntactic)的方法,如 sif,COPS,KOALA,shingling,YAP3,MDR.这类方法都要求从文档中选取一些字符串,这些字符串被称为“指纹”(fingerprint).然后把指纹映射到 Hash 表中,一个指纹对应一个数字.最后统计 Hash 表中相同的指纹数目或者比率,作为文本相似度依据.计算文本相似度的决策函数有很多种,最简单的两种如下:令 $F(A)$表示文档 A 的指纹集,$F(B)$表示文档 B 的指纹集,$S(A,B)$表示文档 A 和 B 的相似度,则第 1 种决策函数为

$$S_1(A,B)=\frac{|F(A)\bigcap F(B)|}{|F(A)\bigcup F(B)|}.$$

第 2 种决策函数为

$$S_2(A,B)=|F(A)\bigcap F(B)|.$$

显然,无论哪种决策函数都有 $S(A,B)=S(B,A)$.另一类文本复制检测采用基于词频统计的方法,这类方法也称为基于语义(semantic)的方法,如 SCAM,CHECK,CDSDG.词频统计法源于信息检索技术中的向量空间模型(vector space model),除此之外,这三者还吸收了信息检索中的其他技术.这类方法首先都要统计每篇文档中各个单词的出现次数,然后根据单词频度构成文档特征向量,最后采用点积、余弦或者类似方式度量两篇文档的特征向量,以此作为文档相似度的依据.

3.1.2 基于字符串比较的方法

Sif,KOALA 和 shingling 的方法基本相同,COPS 略有不同.前三者都需要指定指纹的长度,而后者指纹长度不固定.Sif[17]首先构造一个字符串集,集中每一个元素称为锚(anchor).然后选取一个锚,并从锚之后取 50 个字节的字符作为一个指纹.KOALA[24]不需要锚集,但是需要首先确定一个α值,然后从文档中选取一些长度为α的字符串(KOALA 认为 30~45 字符比较合适,20 个字符最佳)作为文档指纹.Shingling 方法[25]是把 w 个连续的单词称为一个 shingle,然后从文档中选取一定量的 w(在 shingle 方法的系统中,w 为 10 个单词)长的 shingle 构成文档指纹集.COPS[18]并不限定指纹的长度,而是以文档中的一个句子作为一个指纹.但是,如何界定一个句子是否结束,却存在问题.因为句号(.)不仅出现在句子结束处,也出现在缩写词之后,比如“e.g.”(等).这样就会造成一些极短的指纹.所以,为了提高精度,COPS 去除了指纹集中的短单词和短句子.并且,COPS 中一篇文档的指纹之间有重叠,也能提高精度,但是显然增加了索引空间.

YAP3 与 MDR 的方法类似.二者都是通过字符串匹配算法直接在两篇文档中搜寻最大匹配字符串,然后统计匹配字符串作为相似度依据.YAP3[6]使用 RKR-GST(running-karp-rabin greedy-string-tiling)——一种贪婪式字符串匹配算法寻找两篇文档中的最大匹配字符串.MDR[27]首先把候选文档构造成一棵后缀树(suffix tree),然后运用匹配统计算法(matching statistics)直接在被检测文档中寻找最大匹配字符串.MDR 的后缀树需要很大的存储空间,所以,后来 Monostori 等人又提出用后缀向量(suffix vector)存储后缀树[31].后缀向量是从后缀树导出的有向无环图(DAG)的一种存储方式.后缀向量中只保存节点信息,不保存边索引,边标识从字符串中获取,所以极大地节省了空间.

3.1.3 基于词频统计的方法

SCAM 的方法受到了信息检索技术的启示.SCAM[19,20]首先统计文档中各个单词出现的次数,然后按照信息检索中常用的反向索引存储法(inverted index storage)存储文档与词频信息.最后,SCAM 参照向量空间模型(vector space model)提出了相关频率模型(relative frequency model),用以度量文档相似性.向量空间模型一般采用点积或者余弦公式来度量相似性.而相关频率模型其实是对余弦公式进行了改动,试图提高文件复制检测精度.令 D 表示候选文档,Q 表示待检测(或者查询)文档,$F(D)$表示文档 D 的词频向量,$F(Q)$表示文档 Q 的词频向量,α表示各词的权重向量,则 VSM 用余弦公式计算的相似度 $S_v(D,Q)$为

$$S_v(D,Q)=\frac{\sum_{i=1}^{N}\alpha_i^2\cdot F_i(D)\cdot F_i(Q)}{\sqrt{\sum_{i=1}^{N}\alpha_i^2F_i^2(D)\cdot\sum_{i=1}^{N}\alpha_i^2F_i^2(Q)}}.$$

显然,$S_v(D,Q)=S_v(Q,D)$.RFM 首先定义了一个靠近集(closeness set)$c(D,Q)$,用于选取文档 D 和 Q 中出现频度相近的单词.也就是说,$c(D,Q)$包含的单词是 D 和 Q 中都有的单词,并且满足如下公式:

$$\varepsilon-\left(\frac{F_i(D)}{F_i(Q)}+\frac{F_i(Q)}{F_i(D)}\right)>0.$$

其中,$\varepsilon=(2^+,\infty)$,是一个用户可调的参数.然后要计算 D 对 Q 的子集度或者包含度 $Subset(D,Q)$,

$$Subset(D,Q)=\frac{\sum_{w_i\in c(D,Q)}\alpha_i^2\cdot F_i(D)\cdot F_i(Q)}{\sum_{i=1}^{N}\alpha_i^2F_i^2(D)}.$$

显然,D 对 Q 的包含度与 Q 对 D 的包含度不一样,即 $Subset(D,Q)\neq Subset(Q,D)$.所以,RFM 最终的相似度 $S_r(D,Q)$为 $S_r(D,Q)=\max\{Subset(D,Q),Subset(Q,D)\}$.如果 $S_r(D,Q)>1$,则令 $S_r(D,Q)=1$.现在则有 $S_r(D,Q)=S_r(Q,D)$.用 RFM 方法可以更好地检测子集包含式复制,并且ε越大,表示对两篇文档中共有单词的容忍度越大,但是无关文档的匹配机会也会越大,即正误差(false positives)越大;ε越小,正误差越小,但是检测小程度重合文档的能力也越

小.SCAM 并未确定一个普适的ε值,但是认为ε=2.5 对于网络新闻文章比较合适.

CHECK[26]方法的最大特点是把文档结构信息引入了文本复制检测中.CHECK 需要解析每一篇文档,获得其结构特性(structure characteristic),并存入注册数据库中.CHECK 把一篇文档按照其章、节、段落等组织成一棵文档树.树的根节点就是整篇文档,其他节点是文档的一个片段,父节点内容恰好是其子节点内容之和.然后,运用信息检索技术中关键词提取的方法,根据词频提取整篇文档的关键词.由于 CHECK 原型只检测 Latex 文档,而 Latex 文档中含有格式信息.所以,CHECK 在提取关键词时还采用了一些启发式.比如,CHECK 认为那些斜体和粗体的单词一般都是重要的单词,所以把这些单词都看作关键词,而无论其出现频率有多少.接下来,CHECK 统计各个节点上出现的关键词.节点上的每一个关键词都以其在该节点上的频率比重为相应权重.最后,由此构成的树就成为该文档的结构特性.

CHECK 在比较两篇文档时,按照深度优先比较两篇文档结构特性的相应节点,如果父节点不匹配,则子节点就不必比较.最后,统计匹配节点比率,作为相似度依据.CHECK 根据两个节点关键词向量的相似度来判定节点是否匹配.如果关键词向量相似度大于某个阈值,则认为两节点匹配,否则认为不匹配.CHECK 用点积计算关键词向量相似度,公式如下:

$$S(V_A,V_B)=\frac{\sum_{i=1}^{|R|}x_{A,i}\cdot x_{B,i}}{\sqrt{\sum_{i=1}^{|R|}x_{A,i}^2\cdot\sum_{i=1}^{|R|}x_{B,i}^2}}.$$

其中 V_A,V_B 分别是文档 A,B 的关键词向量;$R=V_A\cup V_B$ 是一个参考向量;$\boldsymbol{x}_A$,$\boldsymbol{x}_B$ 分别是经过归一化的 V_A,V_B;关键词向量归一化公式为

$$x_{A,i}=\begin{cases}0, & \text{如果}a_{R,i}\notin V_A\\ w_{A,j}, & \text{如果}a_{R,i}=a_{A,j}\in V_A\end{cases},$$

其中 $w_{A,j}$ 是关键词 a 在该节点的权重.

CDSDG[32]的方法与 CHECK 方法非常类似.它也是把文档按照章、节、段等不同的粒度组织成一棵结构树,然后与 CHECK 方法一样获得每个节点的关键词向量(CDSDG 称为主题向量)和相应的词频向量.但是,在匹配两个节点时,CDSDG 既需要度量两个节点的语义重叠度,又需要度量结构重叠度.语义重叠度就是词频向量的相似度,不过,CDSDG 并没有采用点积或者余弦公式,而是采用了与 SCAM 一样的度量公式.下面给出 CDSDG 中某一粒度的语义重叠度定义.

数字正文 R 与 S 中粒度为 G 的成分 $M_{m2}^G(R)$ 和 $M_{m1}^G(S)$ 的语义重叠度 OoS 为

$$OoS(M_{m2}^G(R),M_{m1}^G(S))=\frac{\sum_{j=1}^{\left\|\vec{W}_{M,m1}^G(S)\cap\vec{W}_{M,m2}^G(R)\right\|}W_{M,m1}^{G,j}(S)\cdot W_{M,m2}^{G,j}(R)}{\sum_{i=1}^{\left\|\vec{W}_{M,m1}^G(S)\right\|}\left(W_{M,m1}^{G,i}(S)\right)^2}.$$

其中 $\vec{W}_{M,m1}^G(S)$ 和 $\vec{W}_{M,m2}^G(R)$ 分别表示数字正文 S 与 R 中粒度为 G 的成分 $M_{m1}^G(S)$ 和 $M_{m2}^G(R)$ 的词频向量;$W_{M,m1}^{G,i}(S)$ 和 $W_{M,m2}^{G,i}(R)$ 依次为 $\vec{W}_{M,m1}^G(S)$ 和 $\vec{W}_{M,m2}^G(R)$ 中第 i 个关键词在其中出现的频度.

CDSDG 中某一粒度的结构重叠度就是两篇文档中对应节点上相同父节点的比率.其定义如下:数字正文 R 与 S 中粒度为 G 的成分 $M_{m2}^G(R)$ 和 $M_{m1}^G(S)$ 的结构重叠度 OoF 为

$$OoF(M_{m2}^G(R),M_{m1}^G(S))=\frac{\left\|ArcHeads(M_{m2}^G(R))\cap ArcHeads(M_{m1}^G(S))\right\|}{\left\|ArcHeads(M_{m1}^G(S))\right\|},$$

其中 $ArcHeads(X)$表示以结点 X 为弧尾的弧头结点的集合.

最后,CDSDG 按照粒度从大到小的次序逐级比较节点,并且只有当语义重叠度和结构重叠度均小于给定阈值时才认为节点匹配,进入下一级节点.当到达叶子节点时,采用基于语句的穷举比较法,以确定是否真正发生了复制行为.

3.2 文本块问题

当两篇文档进行比较时,检测的基本单位称为文本块(chunk).最粗的块就是把整篇文档作为一个块.但是这样只能检测出完全相同、一字不差的复制文本,无法检测出任何部分复制文本.最细的块就是把一个字符作为一个块.但是对于英文文本而言,任何包含字母表所有字母的两篇文本都是复制文本,这显然也不行.所以,合适的块大小一定要介于二者之间.块到底应该多长,各个系统不一样.KOALA 认为 30~45 字符比较合适,20 个字符最佳.Shingle 方法选择 10 个连续的单词,大约 50~60 个字符.Sif 工具选取连续 50 个字节(因为 sif 还要比较二进制文件,所以以字节为单位,不以字符为单位.).MDR 选取 60 个字符作为块长度.SCAM 则从单个单词、5 个连续的单词、10 个连续的单词、一个句子直到整篇文档都实验了一遍.CHECK 和 CDSDG 都以一个单词作为一个块,但是都根据结构信息在多个粒度上对文档进行检测.

显而易见,块长度(粒度)越小,匹配错误的机会越大.很可能把两篇不相关的文档判定为剽窃.另一方面,块长度(粒度)越大,丢失复制文档的机会就越大.这样就会把很多复制文档漏过去.而 CHECK 和 CDSDG 使用变化的粒度,可以获得更好的精度,但是由于在多个粒度上都要比较,需要较多的检测时间.

在选取文本块时,如何确定文本块的边界,文本块之间是否重叠,各个系统也不相同.COPS 做过的实验显示,重叠文本块的检测精度要高于非重叠文本块的检测精度.但是,重叠文本块需要更多的索引空间.另外,对于非重叠文本块而言,插入或者删除一个单词将改变文本块边界,从而会导致检测精度降低.SCAM 建议了一种哈希断点切块法(Hashed breakpoint chunking),就是为每一个单词计算一个 Hash 值并以此确定文本块边界.而 sif 则使用称为“锚”的字符串作为文本块边界.但是,如何获得一个普适的、与文档主题无关的锚集却并不容易.

Sif,KOALA 和 YAP3 所选取的指纹都可能是从一个单词中间开始的字符串,而其他系统都采用更自然的、以单词为边界的指纹边界.在采用单词指纹边界时,一篇文档最大可能的指纹数目就是单词数;但是在采用字符指纹边界时,最大可能的指纹数目就是整篇文档的字符数目.由此可见,采用单词指纹边界可以减小指纹集.由于 sif 还要检测二进制文件,那里根本就没有单词边界,所以 sif 也无法使用单词指纹边界.

3.3 系统结构

sif 的目的是在大文件系统中检测相似文件(包括二进制文件),YAP3 的主要目的是检测程序复制.这两者对待检测文件都不作预处理,也不使用数据库注册(存储)已有文档.其他系统都是面向文本复制检测的系统.它们的系统结构大同小异,都与最早 COPS 系统有类似之处.首先,系统有一个后台大型数据库,存储所有已知的文本数据.其次,系统有一个输入模块,负责搜集或者输入待检测文档,并对原始文档进行清洗和初步处理.一般由输入模块输出的数据都是经过格式转换、分词、词干处理、高频词去除等步骤之后的单词序列.再次,就是比较模块,该模块负责把待检测文档生成的单词序列与数据库中已知的文本数据进行比较,并给出相似性度量.最后,是判定和解释模块,输出检测结果并给出相关解释.

各个系统原型所能检测的原始输入文本格式不尽相同,但是最终都转换为 ASCII 文本.COPS 原型系统检测的文本格式有 Tex(包括 Latex),DVI,troff 和 ASCII 文本.SCAM 原型可以检测 HTML,ASCII 和 postscript 格式的文件.Shingle 方法的系统只用 HTML 和 txt 文本文件做了实验.KOALA 原型只检测 postscript 文件.MDR 原型专门有一个转换部件,负责把常见的文档类型(例如 PS,PDF,MS WORD,HTML)转换成 ASCII 文件.MDR 中原始 ASCII 文件也要转换,以去掉多余的空白符.CHECK 原型只检测 Latex 文档.CDSDG 原型系统只用 HTML 和 txt 文件做了实验.

3.4 总 结

我们把目前已知各种文本复制检测系统的主要内容按照发布时间简单地列在表 1 中.

Table 1 Summary of text copy detection system
表 1 文本复制检测系统汇总

System (Method)	Developer (s)	Year	Feature extracting method	Algorithm of similarity	Text chunk
Sif	U. Manber	1993	String matching	Number of common fingerprints	50 bytes after anchor
COPS	S. Brin, H. Garcia-Molina, *et al*.	1995	String matching	Matching ratio of fingerprints	Sentence
SCAM	N. Shivakumar, H. Garcia-Molina	1995	Word frequency	RFM	Word
YAP3	M.J. Wise	1996	RKR-GST, the longest matching string	Matching ratio	
KOALA	N. Heintze	1996	String matching	Matching ratio of fingerprints	20 characters
CHECK	A. Si, H.V. Leong, *et al*.	1997	SC and key words	Cosine function and matching ratio of section	Word and variable granularity
Shingling	A.Z. Broder, *et al*.	1997	String matching	Matching ratio of fingerprints	10 words
MDR	K. Monostori, *et al*.	2000	Suffix tree, the longest matching string	Matching ratio	60 characters
CDSDG	Song Qin-Bao, *et al*.	2001	SC and key words	Overlap of semantic and overlap of structure	Word and variable granularity

4 结 论

复制检测技术在知识产权保护和信息检索中有着重要的应用.这项技术从 20 世纪 70 年代开始研究、发展.初期的复制检测技术只用简单的属性计数法检测程序复制,现在的程序复制检测都需要综合属性计数和结构度量两方面的信息.文本复制检测略难于程序复制检测,所以,文本复制检测技术直到 20 世纪 90 年代才出现.文本复制检测有两类基本的检测方法,一类是基于字符串比较的方法,另一类是基于词频统计的方法.文本复制检测可以借鉴信息检索中的很多方法,比如 VSM,IIS 和基于结构的文本相似性等等.在文本复制检测中引入结构信息,以实现多粒度检测是提高检测精度的一个重要手段.将来的文本复制检测方法应该和程序复制检测方法一样,综合特征向量和结构度量两方面信息去度量文本相似性.到目前为止,复制检测技术主要集中在文档复制检测上,针对图像、音频、视频的复制检测还有赖于基于内容的检索技术更进一步的发展.

References:

[1] Popek G. J., Kline C. S. Encryption and secure computer networks. ACM Computing Surveys, 1979,11(4):331~356.

[2] Griswold GN. A method for protecting copyright on networks. In: Proceedings of the Joint Harvard MIT Workshop on Technology Strategies for Protecting Intellectual Property in the Networked Multimedia Environment. Cambridge: MIT Press, 1993. 214~221.

[3] Brassil J, Low S, Maxemchuk N, O' Gorman L. Electronic marking and identification techniques to discourage document copying. IEEE Journal on Selected Areas in Communications, 1995,13(8):1495~1504.

[4] Choudhury A, Maxemchuk N, Paul S, Schulzrinne H. Copyright protection for electronic publishing over computer networks. IEEE Network, 1995,9(3):12~21.

[5] Kahng AB, Kirovski D, Mantik S, Potkonjak M, Wong JL. Copy detection for intellectual property protection of VLSI design. In: Proceedings of the Conference on Computer-Aided Design. 1999. 600~604. http://ieeexplore.ieee.org.

[6] Wise MJ. YAP3: Improved detection of similarities in computer programs and other texts. In: Proceedings of the SIGCSE'96. 1996, 130~134. http://citeseer.nj.nec.com/wise96yap.html.

[7] Yoshitaka A, Ichikawa T. A survey on content-based retrieval for multimedia databases. IEEE Transactions on Knowledge and Data Engineering, 1999,11(1):81~93.

[8] Idris F, Panchanathan S. Review of image and video indexing techniques. Journal of Visual Communication and Image Representation, 1997,8(2):146~166.

[9] Lu HQ, Kong WX, Liao Ming, Ma SD. A review of content-based parsing and retrieving for image and video. Acta Automatica Sinica, 2001,27(1):56~70 (in Chinese with English abstract).

[10] Verco KL, Wise MJ. Software for detecting suspected plagiarism: comparing structure and attribute-counting systems. In: Proceedings of the 1st Australian Conference on Computer Science Education. 1996. 3~5. http://citeseer.nj.nec.com/verco96software.html.

[11] Grier S. A tool that detects plagiarism in PASCAL programs. SIGCSE Bulletin, 1981,13(1):15~20.

[12] Gitchell D, Tran N. Sim: A utility for detecting similarity in computer programs. In: Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education. ACM Press, 1999. 266~270. http://doi.acm.org/10.1145/299649.299783.

[13] Prechelt L, Malpohl G, Philippsen M. Finding plagiarism among a set of programs with Jplag. Journal of Universal Computer Science, 2002,8(11):1016~1038.
[14] Parker A, Hamblen JO. Computer algorithms for plagiarism detection. IEEE Transactions on Education, 1989,32(2):94~99.
[15] Clough P. Plagiarism in natural and programming languages: An overview of current tools and technologies. Research Memoranda: CS-00-05, Department of Computer Science, University of Sheffield, 2000.
[16] Singhe S, Tweedie FJ. Neural networks and disputed authorship: New challenges. In: Proceedings of the 4th International Conference on Artificial Neural Networks. IEEE, 1995. 24~28. http://ieeexplore.ieee.org.
[17] Manber U. Finding similar files in a large file system. In: Proceedings of the Winter USENIX Conference. 1994. 1~10. ttp://manber.com/publications.html.
[18] Brin S, Davis J, Garcia-Molina H. Copy detection mechanisms for digital documents. In: Proceedings of the ACM SIGMOD Annual Conference. 1995. http://www-db.stanford.edu/pub/brin/1995/copy.ps.
[19] Shivakumar N, Garcia-Molina H. SCAM: A copy detection mechanism for digital documents. In: Proceedings of the 2nd International Conference in Theory and Practice of Digital Libraries (DL'95). 1995. http://www-db.stanford.edu/~shiva/publns.html.
[20] Shivakumar N, Garcia-Molina H. Building a scalable and accurate copy detection mechanism. In: Proceedings of the 1st ACM Conference on Digital Libraries (DL'96). 1996. http://www-db.stanford.edu/~shiva/publns.html.
[21] Salton G. The state of retrieval system evaluation. Information Processing & Management, 1992,28(4):441~449.
[22] Garcia-Molina H, Gravano L, Shivakumar N. dSCAM: Finding document copies across multiple databases. In: Proceedings of the 4th International Conference on Parallel and Distributed Systems (PDIS'96). 1996. http://www-db.stanford.edu/~shiva/publns.html.
[23] Shivakumar N, Garcia-Molina H. Finding near-replicas of documents on the web. In: Proceedings of the Workshop on Web Databases (WebDB'98) Held in Conjunction with EDBT'98. 1998. http://www-db.stanford.edu/~shiva/publns.html.
[24] Heintze N. Scalable document fingerprinting. In: Proceedings of the 2nd USENIX Workshop on Electronic Commerce. 1996. http://www.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html.
[25] Broder AZ, Glassman SC, Manasse MS. Syntactic clustering of the Web. In: Proceedings of the 6th International Web Conference. 1997. http://gatekeeper.research.compaq.com/pub/DEC/SRC/technical-notes/SRC-1997-015-html/.
[26] Si A, Leong HV, Lau RWH. CHECK: A document plagiarism detection system. In: Proceedings of the ACM Symposium for Applied Computing. 1997. 70~77. http://www.acm.org/pubs/citations/proceedings/ sac/331697/p70-si/.
[27] Monostori K, Zaslavsky A, Schmidt H. MatchDetectReveal: Finding overlapping and similar digital documents. In: Proceedings of the Information Resources Management Association International Conference (IRMA2000). 2000. http://www.csse.monash.edu.au/projects/MDR/papers/.
[28] Monostori K, Zaslavsky A, Schmidt H. Parallel overlap and similarity detection in semi-structured document collections. In: Proceedings of the 6th Annual Australasian Conference on Parallel And Real-Time Systems (PART'99). 1999. http://www.csse.monash.edu.au/projects/MDR/papers/.
[29] Monostori K, Zaslavsky A, Schmidt H. Document overlap detection system for distributed digital libraries. In: Proceedings of the ACM Digital Libraries 2000 (DL2000). 2000. http://www.csse.monash.edu.au/projects/MDR/papers/.
[30] Monostori K, Zaslavsky A, Schmidt H. Parallel and distributed overlap detection on the Web. In: Proceedings of the Workshop on Applied Parallel Computing (PARA2000). 2000. http://www.csse.monash.edu.au/projects/MDR/papers/.
[31] Monostori K, Zaslavsky A, Vajk I. Suffix vector: A space-efficient representation of a suffix tree. Technical Report, 2001.
[32] Song QB, Shen JY. On illegal coping and distributing detection mechanism for digital goods. Journal of Computer Research and Development, 2001,38(1):121~125 (in Chinese with English abstract).
[33] Glatt plagiarism screening program. 2003. http://www.plagiarism.com/screen.id.htm.
[34] Plagiarism.org. 2003. http://www.plagiarism.org.
[35] http://www.canexus.com/eve/abouteve.shtml. 2003.
[36] http://www.wordchecksystems.com/. 2003.
[37] Measure of software similarity. 2003. http://www.cs.berkeley.edu/~moss/general/moss.html.
[38] Bull J, Collins C, Coughlin E, Sharp D. Technical review of plagiarism detection software report. http://www.jisc.ac.uk/. 2003.
[39] Condron F. Plagiarism and the Internet. Report on the Electronic Plagiarism Detection Workshop, JISC (Joint Information Systems Committee). 2001. http://www.oucs.ox.ac.uk/ltg/reports/plag.shtml.
[40] Culwin F, Lancaster T. A review of electronic services for plagiarism detection in student submissions. In: Proceedings of the LTSN-ICS conference 2000. 2000. http://www.ics.ltsn.ac.uk/pub/conf2000/Papers/Culwin.pdf.

附中文参考文献:

[9] 卢汉清,孔维新,廖明,马颂德.基于内容的视频信号与图像库检索中的图像技术.自动化学报,2001,27(1):56~70.
[32] 宋擒豹,沈钧毅.数字商品非法复制和扩散的监测机制.计算机研究与发展,2001,38(1):121~125.