**Computer Science and Application** 计算机科学与应用, 2020, 10(2), 371-378
Published Online February 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.102038

Hans汉斯

# Similar Document Recognition Technology Based on the Improved Simhash Algorithm

**Xinglan Zhang, Dandan He**

Beijing University of Technology, Beijing
Email: 1011684885@qq.com

Received: Feb. 6th, 2020; accepted: Feb. 18th, 2020; published: Feb. 25th, 2020

## Abstract

**[Purpose/Significance]: In order to achieve more efficient in mass text accurately detect the similar text. [Method]: This paper based on Simhash algorithm similar document identification technology improvement, research on Simhash signature value calculation method to make improvements, participle stage using ICTCLAS segmentation system, the text of key method to calculate the weights of the TF-IDF technology, at the same time, the key parts of speech, word length, whether marked word and are included in the title of several major aspects as weighting factor. Finally, the hamming distance is used to compare the document signature value, and the similar documents can be accurately found from the mass documents. [Conclusion]: By improving the TF-IDF weight, the improved Simhash algorithm is better than other algorithms in the recognition accuracy of similar documents.**

## Keywords

**Similar Document Detection, Simhash Algorithm, TF-IDF Algorithm, Fingerprint Calculation, Hamming Distance**

# 基于改进的Simhash算法的相似文档识别技术

**张兴兰，何丹丹**

北京工业大学，北京
Email: 1011684885@qq.com

收稿日期：2020年2月6日；录用日期：2020年2月18日；发布日期：2020年2月25日

## 摘　要

**[目的/意义]：为了实现在海量文本中更加高效准确检测出相似文本。[方法]：本文对基于Simhash算法**

文章引用: 张兴兰, 何丹丹. 基于改进的 Simhash 算法的相似文档识别技术[J]. 计算机科学与应用, 2020, 10(2): 371-378. DOI: 10.12677/csa.2020.102038

**的相似文档识别技术进行研究改进，对Simhash签名值的计算方法作出改进，分词阶段使用ICTCLAS分词系统，文本特征词的权重计算方法采用TF-IDF技术，同时将特征词的词性、词长、是否为标志词与是否被包含在标题中几大方面作为权重计算的考虑因素。最后使用汉明距离对文档签名值进行比较，从海量文档中精确地找出相似文档。[结论]：通过改进TF-IDF权重，使得改进的Simhash算法在相似文档识别准确率上优于其他算法。**

## 关键词

**相似文档检测，Simhash算法，TF-IDF算法，指纹计算，汉明距离**



## 1. 引言

当今时代，互联网的飞速发展，越来越多的数据量给人们带来了非常大的困扰，但是经研究发现，在海量数据中，有大量的数据是相似甚至重复冗余的，并且随着数据的增长，冗余数据变得越来越多，缓解数据中心存储容量已成为巨大挑战[1]。

因此，相似文档识别技术在诸多领域发挥着不可或缺的作用，通过识别出相似甚至重复数据来进行重复数据去重节省空间，使相同的数据只被存储一次，从而解决海量重复文本难题。相似文本识别技术是一种基于内容的检索技术，文本的相似性可以基于文本主题或文字的表达形式来定义。如今在海量文本中，存在大量的非结构化知识文档，当用户输入关键词进行搜索时，使用现有的相似文本识别技术只能匹配搜索到包含该关键词的文档，而一些语义相似的文档未能被搜索出来[2]。

针对相似文本识别不完全以及不准确的问题，本文对传统的 Simhash 算法进行改进，主要是对 Simhash 签名值的计算公式进行改进，在特征词权重计算阶段考虑了更多的影响因素，将关键词的词性、词长、标志词以及标题中是否含有该特征词几大影响因素考虑进去，提出了新的综合加权公式，使 Simhash 签名值更加精确，更能代表该篇文档的主题含义。最后通过仿真实验证明，改进方案在准确度以及执行效率上都有所提升。

## 2. 相关技术

### 2.1. 哈希算法

传统的哈希算法只负责将原始内容尽量随机、均匀地映射为一个唯一的签名值，与此同时带来的问题是即使内容相似或者仅有极少数改动的文档，计算出的哈希值也会相差很大，因此无法使用传统的哈希算法来计算文档的签名值，无法通过这种签名值来判断文档之间是否具有相似性。

### 2.2. Simhash 算法

Simhash 算法是一种缩减维度的算法，旨在将高维的向量用较低维度的签名来表示，是解决相似文本检测的高效哈希技术[3]，它除了能够识别出原文本是否相似之外，还能够识别出不同内容在比特位上的差异程度，Simhash 在海量数据相似文本检测中有大量应用，通过 Simhash 算法生成文档签名值来代表该文档，通过比较文档之间的海明距离来判断签名值之间的相似程度，以此距离来确定网页是否相似。

然而传统的Simhash签名值主要有两个问题[4]：1) Hash值的计算问题，提取特征词速度与精度不高，导致计算出的Simhash签名值无法代表该文档的核心主题。2) 特征词权重计算问题，权重计算公式影响因素考虑不够全面，导致精度丢失，最终也会影响该签名值无法代表文档的核心主题。而在面对海量相似文本检测时，需要Simhash签名值精确的代表该文档的核心含义，以确保相似文本检测的精确性，因此需要对Simhash算法进行优化。

### 2.3. TF-IDF 模型

TF-IDF算法认为某个特征词的权重与在该篇文档中出现的频率成正比，与该特征词在整个文档集中出现的频率成反比[5] [6]。也就是说如果某个字、词在一篇文档中出现的次数很多，并且该字、词在文档集中的其它文档中出现的频率很低，则可以认为该字、词具有良好的特征描述能力。但是，如果某个词不仅在该篇文档中反复出现，在海量文本集中同样反复出现，比如“的”“吗”等一些副词及语气助词[7]，则说明该词不能良好的描述该篇文档的特征。其计算公式可描述为：

$$\mathrm{Weight}(\mathrm{word}) = \mathrm{TF}(\mathrm{word}) * \mathrm{IDF}(\mathrm{word})$$

其中TF即该特征词的词频，表示该特征词对一篇文档的重要程度，其计算公式为[8]：

$$\mathrm{TF}(\mathrm{word}) = \frac{\mathrm{Count}(\mathrm{word},\mathrm{doc})}{\sum_{i=0}^{n}\mathrm{Count}(\mathrm{word}_i)} = \frac{\text{特征词word在本篇文档doc中出现的次数}}{\text{本篇文档共包含的特征词个数}}$$

IDF即逆向文档频率，是指某个特征词word在文档集docs中出现的频率，表示某个特征词在文档集中的分布特征，其计算公式为：

$$\mathrm{IDF}(\mathrm{word}) = \log\left(\frac{\mathrm{Count}(\mathrm{docs})}{\mathrm{Count}(\mathrm{word},\mathrm{docs})} + 0.01\right)。$$

## 3. Simhash 算法的改进

本文方案的优化主要针对Simhash签名值的计算阶段，在第一步文档分词阶段使用ICTCLAS分词系统，选出具有代表性的特征词，使得计算出的Simhash签名值能更好的代表该文档的主题；在权重计算过程中，考虑特征词的词性、词长，给予词长较长的特征词更高的权重，因为我们认为词长越长的特征词对该文档的主题有更好的代表意义，以及“总之”“综上所述”等标志词、被包含在标题中的特征词给与更高的权重，综合以上多方面因素来提高Simhash签名值的精确度。

经过实验验证，本文方案在识别相似文档的准确率以及算法运行时间上都有很大优势。

### 3.1. Simhash 算法的分词阶段

Simhash算法中的对文档的分词技术使用ICTCLAS分词系统，此ICTCLAS分词系统可以快速进行中文分词、标出关键词词性、过滤掉不常用词，分出关键词同时支持用户自定义词典，这些功能为权重的计算提供了很好的基础条件，不仅使数据预处理的精度得到了提升，而且数据预处理速度同样得到了保证。

下文介绍Simhash算法中特征词权重计算阶段，该阶段使用改进的TF-IDF模型计算文档特征词的权重。

### 3.2. TF-IDF 算法改进

本文主要针对TF-IDF算法在特征词权重影响因素考虑不足的问题，在词性、词长、标志词以及文档

标题中是否含有特征词几大方面来对 TF-IDF 算法的权重计算进行改进。

### 3.2.1. 引入词性、词长自适应权重

正如我们所知，文档都是由多个句子组成的，而在句子中主语、谓语是最能代表该句子核心含义的组成部分，主语是执行该句子行为的主体，谓语用来形容该句子主语此时的状态，因此一个句子中主语和谓语应该给予更高的权重，通常情况下，一个句子中的主语大多由名词来承担，谓语大多是动词，因此对于词性不同的特征词应当给予不同的权重。

特征词词性权重的计算公式为：

$$\mathrm{WeightCX}\left(\mathrm{word}_i\right)=\begin{cases}3, \mathrm{word}_i \in N\\ 2, \mathrm{word}_i \in V\\ 1, \mathrm{word}_i \text{为Other}\end{cases}$$

对于特征词的词长，通过 2008 年度 CSSCI 关键词库[9] [10]中的关键词的统计结果得知，特征词词长为 4~6 的占比较多[11] [12]，同时这些词长较长的特征词包含的语义也相对于词长较短的特征词更丰富，能提供更多的文档主题信息，因此应当赋予四个字以及以上的词更高的权重，但是并不能简单的认为长度为 6 的特征词包含的语义信息为长度为 3 的特征词的两倍，所以需要对特征词词长进行归一化处理。

$$\mathrm{Len}\left(\mathrm{word}_i\right)=\frac{\mathrm{Len}\left(\mathrm{word}_i\right)-\mathrm{Len}\left(\mathrm{word}_{\min}\right)}{\mathrm{Len}\left(\mathrm{word}_{\max}\right)-\mathrm{Len}\left(\mathrm{word}_{\min}\right)}$$

其中，$\mathrm{Len}\left(\mathrm{word}_i\right)$为该特征词词长，$\mathrm{Len}\left(\mathrm{word}_{\min}\right)$为所有分词中词长最短的特征词，$\mathrm{Len}\left(\mathrm{word}_{\max}\right)$为所有特征词中词长最长的特征词。

### 3.2.2. 引入标志词的自适应权重

所谓标志词，就是“综上所述”、“总之”“但是”等带有总结或转折意味的标志性的词或短语。在中文表达中，常常用标志词来标识文章中的重要句子，凸显文档主题，现在标志词仍然在文摘系统中受到高度重视。在本文中构建了一个中文标识词词库，并且引入了带有标志词的特征词的自适应权重。

$$\mathrm{Logo}\left(\mathrm{word}_i\right)=lg_i*5$$

其中 $lg_i$ 是一个布尔型变量，用来表示特征词中是否含有标志词。公式中默认含有标志词的特征词的权重为普通特征词的 5 倍。

$$lg_i=\begin{cases}1, & \text{该特征词中含有标志词}\\ 0, & \text{该特征词中不含有标志词}\end{cases}。$$

### 3.2.3. 引入特征词－标题的自适应权重

在中文表达中，标题或者标题中的某个词语会反复出现在文档中，往往含有强调文档主题的意味，因此，在特征词权重计算中，我们对于出现在标题中的候选特征词给予更高的权重，这些特征词往往更能代表本文档的主题，对该篇文档主题的提取有非常大的意义。所以，本文将特征词是否出现在标题中作为权重的考量因素。

$$\mathrm{Title}\left(\mathrm{word}_i\right)=F\_T_i*5$$

其中 $F\_T_i$ 是一个布尔型变量，用来表示特征词是否出现在文档标题中。公式中默认出现在标题中的特征词的权重为普通特征词的 5 倍。

$$F\_T_i = \begin{cases} 1, & \text{该特征词出现在文档标题中} \\ 0, & \text{该特征词未出现在文档标题中} \end{cases}$$

根据本文以上内容，综合加权对于文档中任何一个特征词 $W_i$ 得出一个综合加权公式：

$$\begin{aligned} \text{Weight\_Score}_i = {} & \text{TF}(W_i) * \text{IDF}(W_i) * (1 + \text{WeightCX}(\text{word}_i) \\ & + \text{Len}(\text{word}_i) + \text{Logo}(\text{word}_i) + \text{Title}(\text{word}_i)) \end{aligned}$$

综上所述，通过考虑影响 Simhash 签名值准确度的各种因素[13]，对特征词权重计算方法进行改进，使得 Simhash 签名值精度更高，对相似文档的精确识别打下良好的基础。

### 3.3. 改进的 Simhash 算法设计

1. 文档分词处理：对文档的内容进行分词、去除停用词等一系列处理[14] [15]；

2. 计算特征词权重：为每个特征词( $\text{Term}_i$ )计算相对于该篇文章的 TF-IDF 值作为该特征词的权重( $\text{weight}_i$ )，其中 TF-IDF 值的计算应用上文中改进的 TF-IDF 算法；

3. 计算特征词的 hash 值：对每个特征词使用同一个 hash 函数计算出一个指纹值 $\text{hash}_i$，指纹长度为 $N$ (如 $N = 64$)，其中 $i = 1,2,3,\cdots,n$；

4. 加权：根据步骤 3 产生的 $\text{hash}_i$ 生成加权数字串，对每个特征词的 $\text{hash}_i$ 中的每一位进行处理，若某一位 $\text{hash}_{ij}$ 值为 1，则令此位 $\text{hash}_{ij}$ 值为 $\text{weight}_i$，若 $\text{hash}_{ij}$ 值为 0，则令此位 $\text{hash}_{ij}$ 的值为 $-\text{weight}_i$，其中 $j = 1,2,3,\cdots,64$；(如 $\text{hash}_{11}$ 表示第一个词的第一位)；

$$f\left(\text{hash}_{ij}\right) = \begin{cases} \text{weight}_i, \text{hash}_{ij} = 1 \\ -\text{weight}_i, \text{hash}_{ij} = 0 \end{cases}$$

5. 合并：将步骤 2 中的每个特征词( $\text{Term}_i$ )的加权数字串进行叠加，使其变成一个序列串 $T_j$，其中 $j = 1,2,3,\cdots,64$；

$$T_j = \sum_{i=1}^{n} f_{w_i}\left(\text{hash}_{ij}\right)$$

6. 降维：形成最终的 Simhash 指纹 S，查看步骤 5 中的序列串 $T_j$，其中 $j = 1,2,3,\cdots,64$，如果 $j$ 位上的数值大于 0，那么 S 的第 $j$ 为设为 1，如果 $j$ 位上的数值小于 0，那么 S 的第 $j$ 为设为 0，从而得到该篇文档的 Simhash 签名值。

综合以上 6 个步骤的计算，每篇文档对应一个长度为 $N$ 的 Simhash 签名值，为下文进行相似文档的检测做了良好的铺垫[16]。本文中进行相似文档的检测时，使用的是两篇文档的 Simhash 签名值的海明距离，海明距离越小，则表示两篇文档越相似。

## 4. 实验结果分析

本文使用 Python 语言实现了方案，实验测试在 Windows 环境下进行，操作系统为 Windows10，硬件环境为 Intel(R)Core(TM)i5-8250UCPU@1.60GHz 处理器，8 GB 内存容量，分词系统采用 ICTCLAS3.0。

### 4.1. 实验数据

为了检验本文改进算法对相似数据的检测精度是否有所提高，从互联网下载教育、住房、医疗、金融、交通五个主题的文档各 500 篇，同时使用大量不相关主题的数据进行混淆(约 10000 篇)，来检验相似文档识别的准确度。实验分别运行经典的 Shingle 算法、传统的 Simhash 算法以及本文改进的 Simhash 算

法，观察实验检测结果是否成功检测出各个主题的 500 篇文档。本文采用查准率、召回率以及算法运行时间来进行有效的评价。

$$查准率=\frac{测试结果中真正的相似文档数量}{所有检测为相似文档的数量}。$$

$$召回率=\frac{测试结果中真正的相似文档数量}{所有已知的相似文档数量}。$$

## 4.2. 实验结果

由图 1 和图 2 可知，改进的 Simhash 算法在查准率(约为 95.3%)和召回率(约为 94.0%)上都有较大的提升并且算法相对稳定。

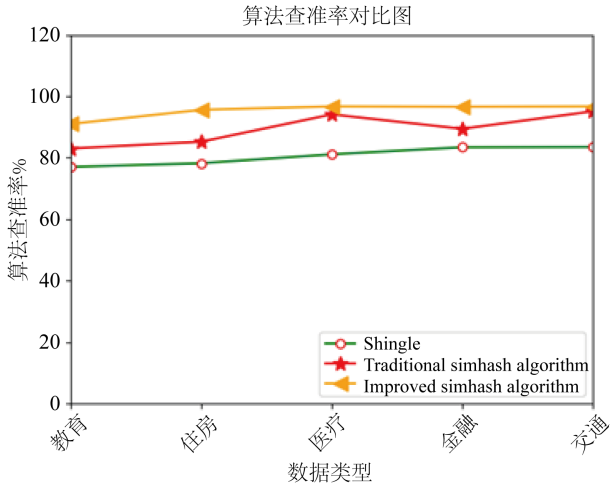


**Figure 1.** Comparison of accuracy of each algorithm
**图 1.** 各算法查准率对比图

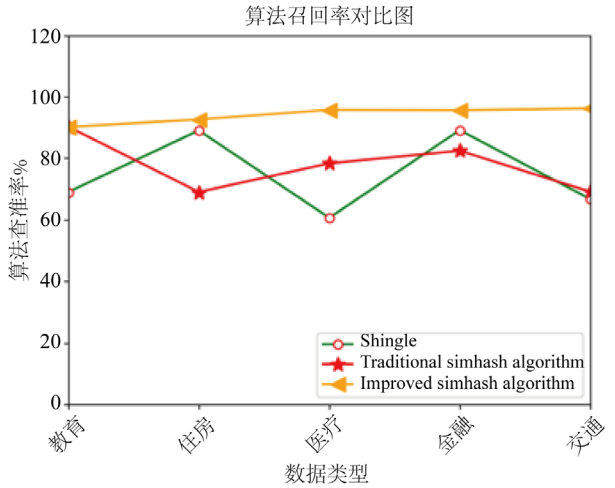


**Figure 2.** Recall rate of each algorithm is compared
**图 2.** 各算法召回率对比图

通过算法查准率对比图可知，改进的 Simhash 算法表现较为稳定，并且总体优于其他两种算法。算法召回率对比图可知，改进的 Simhash 算法稳定性大幅提升，这是因为改进的 Simhash 算法在进行特征

词权重的计算上，对权重公式进行了改进，使用了考虑词性、词长、标志词以及标题词多种因素的综合加权公式，因此对于任何主题的数据，改进的 Simhash 算法都能保持相对稳定且较高的召回率。

实验结果可以看出，Shingle 算法在召回率上表现并不理想，波动较大，这是因为在运行实验的时候对 Shingle 集合采用抽样计算，损失了大量的精度。而传统的 Simhash 算法在权重的计算上考虑因素也过于单一，仅仅考虑了特征词出现的频率，导致签名值的准确率不高，对于不同的主题数据也未能获得较高的召回率。

由图 3 可知，改进的算法在执行时间上明显小于其他两种算法，并且随着文件数据量的增大，执行时间并没有太过明显的变化且明显低于另外两种算法。

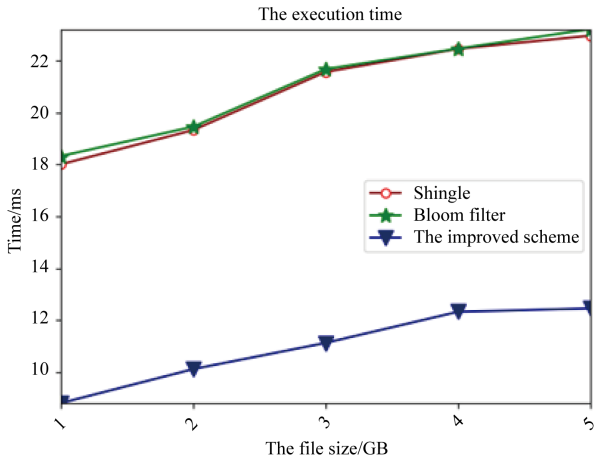


**Figure 3.** Running time of each algorithm is compared
**图 3.** 各算法运行时间对比图

## 5. 结语

相似文本识别技术对当今海量文本去重中具有重大的实用价值，但是目前的相似文本识别技术精度和准确度并没有达到理想的要求，针对这种紧迫的现状，本文对 Simhash 局部哈希算法进行改进，使用了高效的 ICTCLAS 分词系统和改进的 TF-IDF 算法，重新定义了特征词权重的综合计算公式，使得 Simhash 签名值更能代表该篇文档的主题，以提高相似文档的检测精度。

通过以上实验结果表明，在海量文本中进行相似文档的检测时，改进的算法在查准率、召回率以及算法的执行时间上相比于其他算法都有所提升。对于相似文档检测取得的理想结果，未来可以把此项技术应用到海量文本去重技术中，通过仅对检测出的相似文本进行重复性对比，极大地提升了对比速度，具有很好的实用价值。

## 基金项目

国家自然科学基金(61272044，61602019，61801008)，北京市自然科学基金(3182028)。

## 参考文献

[1] 谢平. 存储系统重复数据删除技术研究综述[J]. 计算机科学, 2014, 41(1): 22-30+42.

[2] 任民山, 蔡红霞. 基于 Simhash 算法的海量文本相似性检测方法研究[J]. 计量与测试技术, 2018, 45(4): 78-80.

[3] 陈春玲, 陈琳, 熊晶, 余瀚. 基于 Simhash 算法的重复数据删除技术的研究与改进[J]. 南京邮电大学学报(自然科学版), 2016, 36(3): 85-91.

[4] 林振飞. 基于混合特征的中文文本分类研究[D]: [硕士学位论文]. 沈阳: 东北大学, 2012.

[5] 陈杰, 陈彩, 梁毅. 基于 Word2vec 的文档分类方法[J]. 计算机系统应用, 2017, 26(11): 159-164.

[6] 余意, 张玉柱, 胡自健. 基于 Simhash 算法的大规模文档去重技术研究[J]. 信息通信, 2015(2): 28-29.

[7] 陈琳. 基于存储系统的重复数据删除技术的研究[D]: [硕士学位论文]. 南京: 南京邮电大学, 2016.

[8] 王青松, 葛慧. 相似聚类的二级索引重复数据删除算法[J]. 小型微型计算机系统, 2017, 38(12): 2797-2801.

[9] Bakirass, B.S. (2014) Secure Similar Document Detection with Simhash. Springer International Publishing, New York, 61-75. https://doi.org/10.1007/978-3-319-06811-4_12

[10] 李彬. 基于 Hadoop 框架的 TF-IDF 算法的改进[J]. 微型机与应用, 2012, 31(7): 14-16.

[11] Broder, A.Z. (1997) On the Resemblance and Containment of Documents. Compression and Complexity of Sequences.

[12] 杨旸, 杨书略, 柯闽. 加密云数据下基于 Simhash 的模糊排序搜索方案[J]. 计算机学报, 2017, 40(2): 161-174.

[13] 董博, 郑庆华, 宋凯磊, 等. 基于多 SimHash 指纹的近似文本检测[J]. 小型微型计算机系统, 2011, 32(11): 2152-2157.

[14] 黄承慧, 印鉴, 侯昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法[J].计算机学报, 2011, 34(5): 856-864.

[15] Tu, S.Z. and Huang, M.L. (2016) Mining Microblog User Interests Based on Text Rank with TF-IDF Factor. *The Journal of China Universities of Posts and Telecommunications*, **23**, 40-46. https://doi.org/10.1016/S1005-8885(16)60056-0

[16] 王方伟, 杨少杰, 赵冬梅, 王长广. 基于改进 TF-IDF 的多态蠕虫特征自动提取算法[J/OL]. 华中科技大学学报(自然科学版): 1-6.