计算机工程与应用
*Computer Engineering and Applications*
ISSN 1002-8331,CN 11-2127/TP

# 《计算机工程与应用》网络首发论文

题目： Simhash 算法在文本去重中的应用
作者： 张航，盛志伟，张仕斌，杨敏
网络首发日期： 2019-07-19
引用格式： 张航，盛志伟，张仕斌，杨敏．Simhash 算法在文本去重中的应用．计算机工程与应用. http://kns.cnki.net/kcms/detail/11.2127.TP.20190719.0930.002.html

网络首发时间：2019-07-19 10:56:55
网络首发地址：http://kns.cnki.net/kcms/detail/11.2127.TP.20190719.0930.002.html

# Simhash 算法在文本去重中的应用

张　航，盛志伟，张仕斌，杨　敏

成都信息工程大学 网络空间安全学院，成都 610225

**摘　要：**为了提升 Simhash 算法的文本去重效果、准确率，解决 Simhash 算法无法体现分布信息的缺点，本文提出了基于信息熵加权的 Simhash 算法（简称 E-Simhash）。该算法引入 TF-IDF 和信息熵，通过优化 Simhash 算法中的权重及阈值计算，增加文本分布信息，使得最终生成的指纹更能体现关键信息的比重，并对指纹信息与权重的关联性进行了分析。仿真实验表明：优化权重计算能有效的提升 Simhash 算法的性能，E-Simhash 算法在去重率、召回率、F 值等方面均优于传统 Simhash 算法，并且在文本去重方面取得了良好的效果。

**关键词：**Simhash；信息熵；词频-逆向文件频率；权重优化；文本去重
**文献标志码：**A　**中图分类号：**TP301　doi：10.3778/j.issn.1002-8331.1902-0246

张航，盛志伟，张仕斌，等. Simhash 算法在文本去重中的应用. 计算机工程与应用
ZHANG Hang, SHENG Zhiwei, ZHANG Shibin, et al. Application of simhash algorithm in text deduplication. Computer Engineering and Applications

## Application of Simhash Algorithm in Text Deduplication

ZHANG Hang, SHENG Zhiwei, ZHANG Shibin, YANG Min
School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China

**Abstract:** To improve the text deduplication effect and accuracy of Simhash algorithm, as well as to solve the shortcomings of Sim-hash algorithm that cannot reflect the distribution information, an improved Sim-hash algorithm based on information entropy weighting, abbreviated as (E-Simhash), is proposed in this paper. Firstly, by introducing TF-IDF and information entropy, optimizing the weight and threshold calculation in Sim-hash algorithm, as well as adding the text distribution information, the final generated fingerprint can better embody the proportion of key information. Meanwhile, the correlation between fingerprint information and weight are also be certificated. Finally, the experimental results demonstrate that the performance of Sim-hash algorithm can be effectively improved by optimizing the weight. The modified algorithm is superior to the traditional Sim-hash algorithm in terms of deduplication rate, recall rate and F value, and also has good performance in Chinese similarity detection. Thus, the effectiveness and accuracy of the proposed method is verified.

**Key words:** Simhash; information entropy; team frequency Inverse document frequency; weight optimization; text deduplication

**基金项目：**国家重点研发计划(No.2017YFB0802302)；四川省教育厅项目(No.18ZA0093)；四川省高校科研创新团队项目(No.17TD0009)；四川省学术和技术带头人培养支持经费资助项目(No.2016120080102643)；四川省应用基础项目(No.2017JY0168)；四川省重点研发计划项目(No.2018TJPT0012)；四川省科技支撑计划项目(No.2016FZ0112, No.2018GZ0204)。
**作者简介：**张航（1992-），男，硕士研究生，研究方向为网络与系统安全、大数据安全；盛志伟（1977-），男，硕士，副教授，研究方向为云计算与大数据处理，物联网工程及应用等。

## 1 引言

随着计算机与信息技术的高速发展以及信息存储技术[1]的广泛应用，人们已经步入大数据时代[2]，数字化信息量呈现爆炸式增长。数据量大、复杂度高以及冗余度高是当前大数据信息的特点。研究表明，一些存储系统中的冗余数据已经达到了60%[3]，并且会随着数据量的上升而增多。因此在有限的存储空间和时间内，如何存储更多有效精炼的信息成为当前研究的热点。

在去除冗余数据方面，Simhash算法是当前公认的最好的去重算法。该算法是一种局部敏感哈希算法[4]，它能够将高维数据进行概率降维并映射为位数较少且固定的指纹，之后再对指纹进行相似度比较来反应数据之间的相似程度。其中比较算法通常使用海明距离[5]及编辑距离[6]。Simhash算法优势在于处理速度快，并且结果准确度高。

如今，Simhash算法被广泛应用在近似文本检测、冗余数据去重、异常检测[7]等领域。文献[8]提出了一种基于多Simhash指纹算法，利用多种指纹值经过k维多曲面进行相似度计算，有效地解决了指纹单一，信息丢失严重的问题；文献[9]中在Simhash算法中加入了减值运算，对最后合并的结果序列串结果减去一个阈值T，从而提升了Simhash算法的准确性。文献[10]中将Simhash算法和CNN进行结合用于恶意软件检测，通过转化为灰度图像提高恶意软件识别率和性能。

但是以上对Simhash的应用都存在一些问题，首先他们没有突出关键项在Simhash指纹中的比重，比如文献[8]中只是简单的进行了术语长度统计从而确定文章的信息，文献[9]中设置关键词权重为1，这样造成严重的信息失真。其次他们没有考虑到信息位置分布对指纹的影响。为了提升Simhash算法的文本去重效果、准确率，解决Simhash算法无法体现分布信息的缺点，引入信息熵的概念，采用熵加权的方式给文档中的关键词进行赋权，优化权重计算公式，并在hash计算中加入关键词分布信息，从而达到对传统Simhash算法的优化，最后通过仿真实验论证了该算法的可行性，合理性。

## 2 相关问题研究

### 2.1 Simhash算法的分析

**定义一** Simhash算法的原理是对于两个给定的变量x,y,哈希函数h总是满足下式：

$$\Pr_{h\in F}\big(h(x)=h(y)\big)=\mathrm{sim}(x,y) \quad (1)$$

其中，Pr表示$h(x)=h(y)$的可能性，$\mathrm{sim}(x,y)\in[0,1]$是相似度函数，一般也用雅可比函数$\mathrm{Jac}(x,y)$来表示变量x,y的相似度，$\mathrm{sim}(x,y)$表示如下：

$$\mathrm{sim}(x,y)=\mathrm{Jac}(x,y)=\frac{|x\cap y|}{|x\cup y|} \quad (2)$$

h属于哈希函数簇F，需要满足以下条件：

1）如果$d(x,y)\le d_1$，则$\Pr_{h\in F}\big(h(x)=h(y)\big)\ge p_1$；

2）如果$d(x,y)\ge d_2$，则$\Pr_{h\in F}\big(h(x)=h(y)\big)\le p_2$。

称F为$(d_1,d_2,p_1,p_1)$上的敏感哈希簇函数[11]。其中$d(x,y)$表示x,y变量之间的距离，通俗而言，表示如果x,y足够相似时，那么它们映射为同一hash函数的概率也就足够大，反之哈希值相等的概率足够小。

由于传统hash函数[12]与Simhash函数最大的不同在于局部敏感性，如果针对输入的数据做些局部些许修改，经过传统hash函数运算后可能会得到完全不同的结果，而Simhash计算的结果则很相似，因此可以使用Simhash函数产生的指纹相似程度来表示源数据之间的相似程度。

### 2.2 Simhash算法流程

Simhash算法的流程是首先定义一个f维度的空间，然后在这个空间中定义每一个特征所对应的向量，接着将所有的向量结合自身的权重进行加权、求和就得到了一个和向量作为结果。最后再对该结果进一步的进行压缩转化，其规则是：对每一个向量得出一个相对应的f位签名信息，若

向量维度的值大于 0，则置其签名所在的位置为 1，否则置为 0。通过这样的转化方式，得到的签名信息就表证了此向量在各个维度中的值的信息。Simhash 的算法流程图如图 1 所示：

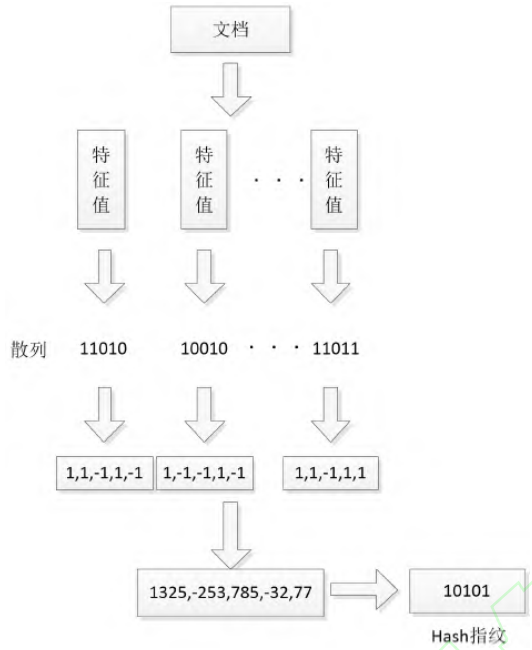


**图 1　Simhash 指纹生成**

Simhash 算法的具体步骤如下：

**步骤 1:**初始化

针对数据集大小及存储成本确定 Simhash 位数以及f维向量空间，同时初始化f位二进制数S均置为 0。

**步骤 2:**文档预处理

主要包含两部分，第一部分是分词，寻找文档的特征词汇以及去除文档停用词等。第二就是赋权，一般而言这里普遍忽略了权重的计算设置为 1[13]。

**步骤 3:**生成 hash 值

利用传统的散列算法对步骤二中的每个特征词计算一个f位 hash 值，并进行下列运算：

for k in V:
　for iinf:
　　if(i == 0):
　　$V_i = V_i + w_{ki}$
　　else:
　　$V_i = V_i - w_{ki}$

**步骤 4:** 压缩变换：

针对最后生成的向量 V，对每一位进行转化处理。

if V[i]>0:
　　S[i]=1
　else:
S[i]=0

**步骤 5:**指纹生成

输出最终的签名 S 作为该文档的指纹，之后再进行海明距离或编辑距离计算相似度。

**步骤 6:**距离计算

在 Simhash 算法中通常使用海明距离进行相似度计算。海明距离通过比较两个文档指纹中不相同的个数来度量两个文档之间的相似度[14]。海明距离越大，代表两个字符串的相似度越低，反之则两个字符串相似度越高。对于二进制字符串而言，可以使用异或运算来计算两个二进制的海明距离。举例说明如下：

**例 1：**设a, b为两个二进制数，其中

$$a = 00110, b = 01110$$

则可知a, b两个二进制数只有第二位不同，故$\mathrm{Hamming}(a, b) = 1$。

也可利用异或操作，统计异或结果中 1 的个数。$a \oplus b = 01000$，共有 1 个 1，故海明距离为 1。

## 2.3　Simhash 存在的一些问题

传统 Simhash 算法在权重计算方面通常设置为 1 或特征词出现的次数，这很容易造成信息丢失，导致最终的 Simhash 指纹准确性降低，并且根据 Simhash 算法可知它不表现出词汇分布信息，关键特征词调整顺后，不会影响最终生成的 Simhash 指纹。

如图 2 所示，两个关键词的位置调整下就可能导致最终的意义大不相同，但是传统的 Simhash 算法生成的指纹却是一样的。

字符串1：能力 比 学历 重要性 高
Simhash指纹：　14826313989210732151
字符串2：学历 比 能力 重要性 高
Simhash指纹：　14826313989210732151

**图 2　位置对 Simhash 的影响**

# 3　改进的 Simhash 算法

本文提出了一种新的基于信息熵的 Simhash 算法，考虑到传统 Simhash 算法中针对权重的计算不充分，以及不能更好的反应文档中词汇的分布特征，本文中引入信息熵理论来解决上述问题，并且在 hash 计算中加入位置关系特征，从而提升 Simhash 算法的准确度。

### 3.1 熵加权权重计算方法

（1）词频-逆向文件频率

词频-逆向文件频率（TF-IDF）[15]算法是一种常用的文本特征权重计算方法，特征词$t_k$在文档$d_j$中的TF-IDF值记为$tfidf(t_k, d_j)$,有如下定义：

定义二 特征项$t_k$在文档$d_j$中出现的频率$tf(t_k, d_j)$为

$$tf(t_k, d_j) = \frac{n_{j,k}}{\sum_i n_{j,i}} \quad (3)$$

式中$n_{j,k}$表示特征词$t_k$在文档$d_j$中出现的次数，$\sum_i n_{j,i}$表示文档$d_j$中的所有特征词的个数。

定义三 反文档频率$idf(t_k, d_j)$是权衡特征词重要性的系数，其定义为：

$$idf(t_k) = \log\frac{|D|}{|\{j:t_k \in d_j\}|+1} \quad (4)$$

式中：$\{j: t_k \in d_j\}$为含有特征词$t_k$的文档综述，$|D|$为语料库中的文件总数。

定义四 TF-IDF 函数,特征词的词频权重定义为：

$$w_k = tfidf(t_k, d_j) = tf(t_k, d_j) * idf(t_k) \quad (5)$$

（2）信息熵

信息熵[16]是由香农在1948年提出的一个概念，用它来表示在随机事件发生之前的结果不确定性的量度，以及在随机事件发生之后，人们从该事件中得到的信息量。

根据信息熵的定义：

$$H(X) = -\sum(x_i \in X)P(x_i)\log_2 P(x_i) \quad (6)$$

其中X表示信息概率空间$X = (x_1: P(x_1), x_2: P(x_2), \dots, x_n: P(x_n))$, $H(X)$表示随机变量X不确定性的量度。

（3）左右信息熵

左右熵[17]是指多字词表达的左边界的熵和右边界的熵。左右熵的公式如下:

$$E_L(W) = -\sum_{\forall a \in A} P(aW|W) * \log_2 P(aW|W) \quad (7)$$

$$E_R(W) = -\sum_{\forall b \in A} P(Wb|W) * \log_2 P(Wb|W) \quad (8)$$

式中，W表示某个单词，$E_L(W)$表示该单词的左熵，$P(aW|W)$表示该单词左侧出现不同词的概率，a变量是一个变化值，表示与W相结合的词汇。$E_R(W)$右熵同理。

（4）熵加权计算法

本文采用熵加权计算方法

$$H_k(w) = \frac{E_l(w)+E_r(w)}{2} \quad (9)$$

这里对特征词左右信息熵取平均。用$H_k(w)$来表示该单词的熵信息量。把熵因子$H_k$加入权值计算公式中，取两者的平方平均数作为词权重，如下所示:

$$etfid(t_k, d_j) = \sqrt{(tfidf(t_k, d_j)^2 + {H_k}^2)/2} \quad (10)$$

上式的物理意义为:特征项$t_k$在文档$d_j$中出现的次数越多，在训练集中出现该特征项的文档越少，并且其信息量越大，则其权重越高。

### 3.2 基于熵加权的 Simhash 算法

基于信息熵的 Simhash 算法主要是在权重方面进行优化，首先利用基于 TF-IDF 算法与信息熵进行加权得到权重，并按照其在文档中的分布进行排序，针对每个特征词汇生成的 hash 将再与其所在位置进行异或。

但是经过改进的权重计算后，由于训练集的不完整等因素，会导致部分特征次权重过大，最终引起查准率下降，为了解决这一问题，引入权重阈值$W_t$。下面就权重不均导致的问题进行证明。

设一个文档中提取出n个关键词分别为$\{p_1, p_2, p_3, \dots p_n\}$，各关键词的权重为$W = \{w_1, w_2, w_3, \dots, w_n\}$。对n个关键词生成hash值，其结果为$H = \{h_1, h_2, h_3, \dots, h_n\}$，经过叠加后生成二级指纹$F = \{f_1, f_2, f_3, \dots f_m\}$, m为指纹位数，最后根据F中$f_i$是否大于 0 生成Simhash 指纹为S。

则若存在某一特征词$p_k$,其权重

$$w_k \gg w_j, j \in [1, n] \cap j \neq k \quad (11)$$

则S主要由$p_k$决定。证明如下:

设 $h_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{im}\}$, $a_{ij}$ 是一个二进制变量，则

$$f_j = \sum_{i=1}^{n} -(-1)^{a_{ij}} w_i \quad (12)$$

提取出$w_k$，则有

$$f_j = w_k \sum_{i=1}^{n} -(-1)^{a_{ij}} \frac{w_i}{w_k} \quad (13)$$

因$w_k \gg w_j$,故：

$$\frac{w_i}{w_k} \approx 0, i \neq k \quad (14)$$

所以此时：

$$f_j \approx w_k * \left(-(-1)^{a_{kj}} \frac{w_k}{w_k}\right) = -w_k * (-1)^{a_{kj}} \quad (15)$$

最终有F主要与$p_k$相关，证明完成。

以上证明同时也反映出权重对Simhash 指纹的影响。

引入权重阈值后，此时的权重计算如式(13)所示:

$$w_k = \begin{cases} \sqrt{(tfidf(t_k, d_j)^2 + H_k^2)/2} & w_k \le W_t \\ w_k & W_t w_k > W_t \end{cases} \quad (16)$$

综上所述，E-Simhash 算法流程如图 3 所示：

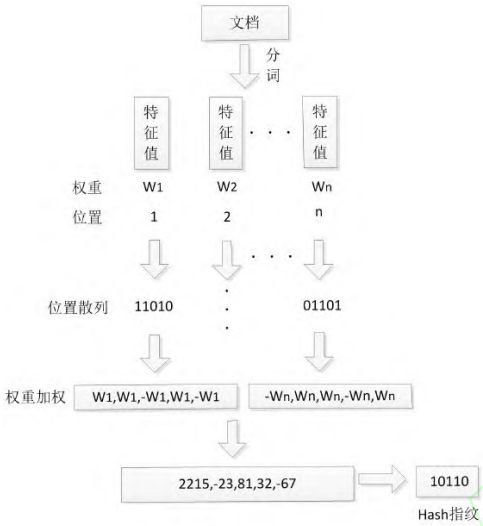


**图3　E-Simhash算法过程**

E-Simhash 算法与传统的 Simhash 算法有以下三点不同，这里主要在 TF-IDF 的基础上引入信息熵进行特征词权重计算，并使用两者的平方平均数作为最后的特征词权重，同时为了避免权重过高的情况导致指纹失真，引入权重阈值,计算方式如式(16)所示。最后在生成特征词 hash 时与特征词位置进行异或，使其 hash 包含文档的位置分布信息。

## 4　仿真实验与分析

本节主要模拟真实的应用场景，验证 E-Simhash 算法的性能是否比传统 Simhash 算法优越。

### 4.1　实验环境及数据集

实验环境部署在一台台式机上，机器参数如表 1 所示：

**表 1　实验环境参数**

| 类别 | 机器型号 | 系统 | 存储容量 | 内存 | 运行环境 |
|---|---|---|---|---|---|
| 实验机 | DELL R530 | Windows 2012 | 1024 GB | 8GB | Python2.7<br>Mysql 5.5.53 |

数据集来自搜狗实验室中的全网新闻数据 2012 版，它是来自多家新闻站点近 20 个栏目的分类新闻，剔除低于 800 字符的数据，并从中随机选取 1565 篇进行后续实验。

首先从 1565 篇新闻中，根据修改比例，随机选取若干篇新闻进行修改、删除、移位、替换等随机操作，并控制修改后的文章与原始文章有一定阈值 T 的相似度，生成待测样本集，之后使用传统 Simhash 与本文中的算法进行比较，统计实验的相关指标。

### 4.2　实验结果分析

实验结果中常用四种指标进行评估，分别是去重率、查准率、召回率以及 F 值[18]，其中去重率是指分类正确的样本数与总样本的比值，就本实验而言即预测为同源文章集数与总文章数的比值。

实验一：去重率对比

在 1565 篇新闻中随机选取 1162 篇进行任意修改，选取不同的海明距离，对比两种算法中的准确率,试验中 T=15%，即每篇新闻保持不超过 15%修改，指纹长度为 128 位，词权重阈值$W_t = 90$，实验结果如图 4 所示：

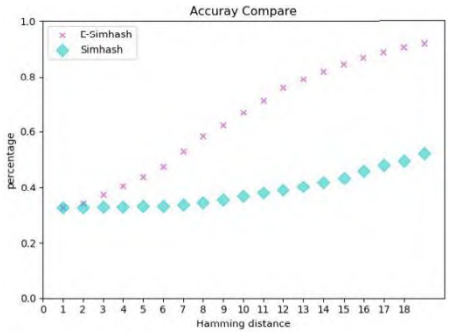


**图 4　不同海明距离下去重率对比**

实验结果表明在海明距离大于 2 时，E-Simhash 算法均具有很高的去重率。在实际应用中海明距离一般取 10 左右，所以 E-Simhash 算法的去重效果更好。

实验二：修改 T 阈值对比

本次实验修改文本的相似度阈值 T，分别对 5%、10%、15%、20%的修改下，海明距离选为 10，即低于 10 则认为相似，比较两种算法的去重率，结果如图 5 所示。

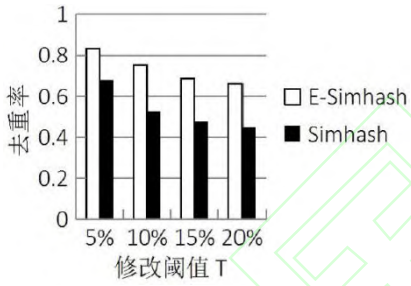


**图 5　不同阈值下去重率对比**

从实验结果中可知，E-Simhash 算法去重率分别以 0.833:0.679、0.751:0.529，0.687:0.476、0.661:0.451 优于传统的 Simhash 算法，并且随着文章变动的增加，其去重率都呈现下降趋势。实验结果表明在不同修改阈值 T 下，E-Simhash 算法均优于传统的 Simhash 算法。

实验三：查准率、召回率以及 F 值对比

在实验中，从新闻集中随机选取一篇文章进行随机修改，并保证与原文有 90%的相似度，对比基于 Simhash 指纹与 E-Simhash 算法的查准率、查全率以及 F1 值。其中海明距离选取 10；实验进行 100 次，并取他们的平均值，作为最终结果，结果图 6 所示：

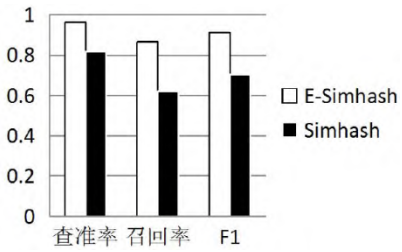


**图 6　综合性能比较**

通过实验数据可知，E-Simhash 算法在查准率 0.963:0.818，召回率 0.867:0.621，F1 值 0.912:0.706 优于传统的 Simhash 算法。结果表明 E-Simhash 算法在查准率、召回率以及 F 值方面均比普通的 Simhash 算法有很大的提升，也足以证明 E-Simhash 算法的优越性。

## 5　总结与展望

针对传统 Simhash 算法在权重计算方面的欠缺，以及算法中不能考虑到文档特征词汇的分布信息，本文通过优化权重计算，使用 TF-IDF 和信息熵的平方平均数作为特征词的权重值，考虑到部分权重过大导致信息失真，引入权重阈值，并在此基础上将特征词的位置信息引入到 hash 计算中去，从而提升 Simhash 算法的去重率、查准率，并通过仿真实验论证了 E-Simhash 算法在各方面均优于传统的 Simhash 算法，但是 E-Simhash 算法依然存在一些不足，在短文本相似度检测方面准确度不高，而且本文中的权重计算方法仍有可改进之处，计算中的关键词权重未必非常准确，未来可通过优化权重计算，如引入 LDA 主题模型[19]可提升 Simhash 算法的适应范围。

## 参考文献：

[1] Bhat, WA. Bridging data-capacity gap in big data storage[J]. Future Generation Computer Systems-the International Journal of Escience, 2017, 87: 538-548.

[2] Chen T. Analysis of Computer Data Processing Mode based on Big Data Era[J]. Agro Food Industry Hi-Tech, 2017, 28(1): 828-831.

[3] Clements A T, Ahmad I, Vilayannur M, et al. Decentralized deduplication in SAN cluster file systems[C]//Conference on Usenix Technical Conference. USENIX Association, 2009.

[4] LEE K M. Locality-sensitive hashing techniques for nearest neighbor search[J]. Interna-

tional Journal of Fuzzy Logic and Intelligent Systems, 2012, 12(4): 300-307.
[5] PARSE B, Othman E. On minimal Hamming compatible distances[J]. RAIRO-Theoretical Informatics and Applications, 2014, 48(5): 495-503.
[6] MARTIN, Ryan R. On the computation of edit distance functions[J]. Discrete Mathematics, 2015, 338(2):291-305.
[7] 周龙泉, 卫文学. 基于主成分分析与 Simhash 的入侵检测方法[J]. 计算机与数字工程, 2015(7): 1291-1294.
[8] 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.
[9] 陈波, 潘永涛, 陈铁明. 基于多层 SimHash 的 Android 恶意应用程序检测方法[J]. 通信学报, 2017(s2):30-36.
[10] NI S, Qian Q, Zhang R. Malware Identification Using Visualization Images and Deep Learning[J]. Computers & Security, 2018:S0167404818303481.
[11] Ma Y, Feng X, Liu Y, et al. BCH–LSH: a new scheme of locality-sensitive hashing[J]. IET Image Processing, 2018, 12(6): 850-855.
[12] Li Q, Sun Z, He R, et al. Deep supervised discrete hashing[C]//Advances in Neural Information Processing Systems, 2017: 2482-2491.
[13] 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(2):161-174.
[14] 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.
[15] 宋人杰, 余通, 陈宇红, 等. 基于 MapReduce 模型的大数据相似重复记录检测算法[J]. 上海交通大学学报, 2018.
[16] SHANNON C E. A mathematical theory of communication[J]. Bell Labs Technical Journal, 1948, 27(4):379-423.
[17] 邢恩军, 赵富强. 基于上下文词频词汇量指标的新词发现方法[J]. 计算机应用与软件, 2016, 33(6):64-67.
[18] 刘震, 陈晶, 郑建宾, et al. 中文短文本聚合模型研究[J]. 软件学报, 2017(10):154-172.
[19] Li Y, Zhou X, Yan S, et al. Design and Implementation of Weibo Sentiment Analysis Based on LDA and Dependency Parsing[J]. China Communications, 2016, 13(11):91-105.