赵天舒, 沈颖, 李柏岩,等. 基于扩展 Trie 树的中文敏感词变体检测[J]. 智能计算机与应用,2024,14(4):215-221. DOI:10.20169/j.issn.2095-2163.240435

基于扩展 Trie 树的中文敏感词变体检测

赵天舒[1,2], 沈 颖[2], 李柏岩[1], 刘晓强[1], 朱 旻[1]
(1 东华大学 计算机科学与技术学院, 上海 201620; 2 上海市计算机软件评测重点实验室, 上海 201112)

摘 要: 网络语言表达方式的随意性和自由性使词语变体在网页上经常出现,给网页信息安全带来了挑战。本文针对中文敏感词变体检测问题,提出一种基于扩展 Trie 树的敏感词变体快速检测方法。首先,对中文敏感词变体类型进行归类,结合中文敏感词特点,通过增强节点内信息和节点间联系构建扩展 Trie 树;再依据中文变体的生成规则检索 Trie 树;最后,使用基于 BERT 的二分类算法对结果进行二次判别,降低误检率。实验表明:该算法精准度达到 98.69%,召回率达到 94.25%,能够识别常见的中文敏感词变体并在时间效率上满足应用需求。

关键词: 敏感词; 词语变体; Trie 树; BERT
中图分类号: TP391.1 **文献标志码**: A **文章编号**: 2095-2163(2024)04-0215-07

Chinese sensitive word variant detection based on extended Trie tree

ZHAO Tianshu[1,2], SHEN Ying[2], LI Baiyan[1], LIU Xiaoqiang[1], ZHU Min[1]
(1 School of Computer Science and Technology, Donghua University, Shanghai 201620, China;
2 Shanghai Key Laboratory of Computer Software Testing and Evaluating, Shanghai 201112, China)

Abstract: The arbitrariness and freedom of expression in internet language often lead to various word variants appearing on web pages, posing a challenge to web information security. In this paper, a fast detection method of sensitive word variants based on extended Trie tree is presented, which can be used to detect Chinese sensitive word variants. This paper first classifies the types of Chinese sensitive word variants, then builds an extended Trie tree by enhancing the information within the nodes and the connections between the nodes, then retrieves the Trie tree according to the generation rules of Chinese variants, and finally uses the BERT-based binary classification algorithm to discriminate the retrieval results twice to reduce the false detection rate. Experiments show that the accuracy of the algorithm is 98.69% and the recall rate is 94.25%. The algorithm can recognize common Chinese sensitive word variants and meet the application requirements in time efficiency.

Key words: sensitive words; word variants; Trie tree; BERT

0 引 言

随着互联网的快速发展和普及,网络信息呈井喷式增长,每天都有大量文本信息发布在网络上,而不法分子常常在其中参杂不良内容,如政治敏感、辱骂、暴力、赌博、毒品、色情等,利用网络非法传播信息。针对这种情况,网络平台通常会使用敏感词过滤系统来屏蔽违规内容。然而,很多不法分子为了逃避这种监管,常常会使用敏感词的各种变体来传达信息。

敏感词变体一般通过对敏感词构成字符,根据音、形、意等特征进行变换而得到,尽管形式上与原词差别很大,仍能传播原来的语义。由于汉字字形多样,发音多变,网络语言构建灵活,所以中文敏感词变体形式种类繁多,但大体仍是从字形、字音角度去变形,遵从字形的拆解的变形或是拼音替换的变形。例如,将“破解”变形为“石皮角刀牛”,将“加微信”变形为“加 weixin”等等,现代网络用语中还习惯用拼音的首字母缩写进行替换,如“yyds”是指“永远的神”,“xswl”是指“秀死我了”等等。显然,敏感

作者简介: 赵天舒(1998-),男,硕士研究生,主要研究方向:自然语言处理,敏感词检测; 沈 颖(1974-),女,硕士,高级工程师,主要研究方向:软件质量,智能检测; 刘晓强(1968-),女,博士,教授,硕士生导师,主要研究方向:智能管理,知识管理; 朱 旻(1999-),女,硕士研究生,主要研究方向:图像处理。
通讯作者: 李柏岩(1968-),男,博士,副教授,硕士生导师,主要研究方向:机器学习,信号处理。Email:libaiyan@ dhu.edu.cn

收稿日期: 2023-04-11

词变体数量极大,无法像敏感词一样通过构建词表来过滤,如何准确、快速在大量的网页文本信息中准确识别敏感词及其各种变体且不干扰正常的文本表达,是一个亟待解决的问题。

敏感词变体的识别大体上可分为基于规则的方法和基于相似性的方法。前者根据敏感词变体产生模式,使用字符串匹配技术识别变体;后者则根据敏感词识别变体。基于相似性方法是非确定性方法,通过计算字符串相似性来识别变体,如神经网络识别方法等。本文采用基于规则的方法,利用汉字音、形的对应关系,通过对基本的 Tire 树节点进行扩展,实现多种变体的快速匹配,能够识别目前网络上常用的几种变体,考虑到本方法在检测敏感词变体时会产生少量的误检问题,本文采用微调训练的 BERT 模型来减少误报率,经测试取得了较好效果。

1 相关研究

早期敏感检测利用敏感词表和字符串匹配技术检索文本中的敏感词,但该方法对敏感词变体无能为力。文献[1]总结某些特殊字符与英文字母相似,通过对特殊字符进行替换,来识别英文敏感词变体,如将“p! ss”还原成“piss”再识别,一定程度上解决了英文变体词的识别问题;文献[2]提出面向 PDF 文本的高效模式串匹配算法,在模式串较长且规模较大时,算法效率较高,但是中文敏感词普遍偏短,不适合使用这种算法进行检索;文献[3]通过构建计算语言资源库,包括词汇表、词根、前缀和后缀等,用于辅助进行敏感词的变体识别;文献[4]针对英文敏感词,从字母语音和字形角度计算变体和原词的相似度,以此来判断是否为敏感词,进一步优化思路,但是仍然只是识别英文的变体。中文方面,文献[5]基于变体识别的敏感词检测方法,提出了包括同义词、替代词、缩略语和屈折词在内的基于编辑距离的相似度计算方法,实现了敏感词变体检测;文献[6]提出了一种基于关联词和扩展规则设计敏感词库的方法,结合广义 Jaccard 系数的相关度扩充敏感词库,再进行匹配识别;文献[7]提出了一种基于关键词的语义相似度计算方法,如果两个词在语义上指向同一个事物或是两个词能组成一个完整的词语,那么这两个词便具有较高的相似度;文献[8]提出基于 k 近邻网络敏感信息过滤方法,引入时间和主题相关度计算相似度,考虑到了网络文本敏感信息稀疏的特点,但针对敏感信息只是计算词向量的余弦相似度,未结合中文变体的特征;文献[9]提出基于音形码的汉字相似度计算方法,通过汉字拼音和字形两部分结合,计算改进的汉明距离判断两个汉字的相似度,但对单一字形相近或发音相近的汉字无法有效识别;文献[10]提出改进音形码与语言知识库 HowNet 相结合计算相似度的方法,在文献[9]的基础上更有效计算音形的相似度;文献[11]通过将敏感词储存在有向图中,利用深度优先搜索对图进行搜索,从而识别敏感词;文献[12]提出将敏感词存入 Trie 树,同时在节点嵌入汉字拼音,使其能识别敏感词以及拼音替换的变体,为敏感词变体识别的数据存储提供了思路,解决了单一拼音替换变体的识别问题,但是对其他变体仍然无法识别;文献[13]提出的 RSWDT(Recognition of Sensitive Words based on Decision Tree)算法进一步在节点上扩充了汉字的区位码,以此来识别部分字形拆解后的汉字,进一步提高了变体检测能力,但是对于拆解后信息有缩略的变体无法识别,例如“破解”的变体“石皮角”,“微信”的变体“wx”,均无法有效识别;文献[14]提出改进的 Trie 树结合深度优先搜索,将拼音独立成单个节点,以识别拼音和汉字混合的变体。基于上述研究,本文通过基于 Trie 树的变体检索方法,能有效识别出中文敏感词的多种变体形式。

2 敏感词变体

网络出现中文敏感词的各种变体,主要依据汉字发音与字形的一些规则构造出来,可分为 4 种类型:

1)特殊字符混入

在敏感词中混入特殊字符的方法有两种:一种是直接在敏感词中插入特殊字符,如“出售手 &! 枪”,以规避敏感词表的直接匹配;另一种是使用特殊字符替换敏感词中的部分字,如变体“卡 * 因”中以“ * ”代替了“洛”字,但不影响整体词义的传达。

2) 拼音替换

用拼音替换汉字也是网络用语常见的构建方法,恶意用户也会利用这点对敏感词进行变形。变形后的敏感词中文和字母夹杂或完全由字母组成,但对阅读效果影响较小,如“全能神”的变体“quan 能神”和“quannengshen”,对于母语是汉语的人来说,容易区分出声母和韵母,因此即使变体全部是拼音字母,也很容易理解传达的含义。

3) 字形拆解替换

汉字是图像化的文字,是一套复杂的符号系统,部分汉字可以根据拆字规则拆解为构字元素序列,

敏感词变体的字形变体通常会使用拆解后的构字元素序列替换原本的汉字，如“破解”的变体“石皮解”就是将“破”字按字形拆成了“石”和“皮”。

4）缩写

变体缩写以原词单字粒度进行。例如，拼音可以缩写为单个首字母；字形拆解后的构字元素序列可以缩写为序列的某一段。一般缩写后的词不会影响语义的表达，如下图 1 所示。

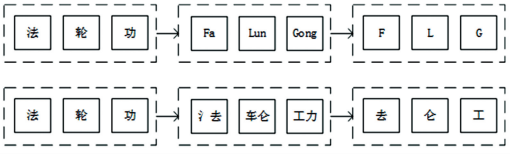

图 1　缩写变体

Fig. 1　Abbreviation variant

以上 4 种变体的粒度是单个汉字，但对某些敏感词来说，一个变体可能由多种变换规则生成的混合变体。如图 2 所示，其既包含了拼音替换和字形拆解替换，也包含了缩写变体。

图 2　混合变体

Fig. 2　Mixed variant

3　敏感词变体检测算法

3.1　基于扩展 Trie 树的敏感词及其变体检索

Trie 树，又称“字典树”，是树形的数据结构，能实现数据高效存储和多模式匹配。Trie 树的插入和查询时间复杂度都为 $O(k)$，其中 k 为模式串的最大长度。Trie 的核心思想是以空间换时间，利用公共前缀来减少查询时间，常被用于字符串的快速检索，候选词推荐，模糊匹配等。Trie 树也常被看成一种确定有限状态机（DFA），每个节点代表一个状态，AC（Aho-Corasick，AC）自动机是基于 Trie 树的多模式串匹配算法[15]。

本文利用 Trie 树作为基本数据结构，并对其进行扩展，存储敏感词及其相关信息，同时结合 AC 自动机，支持多种敏感词变体查询，扩展主要集中在以下 3 点：

（1）扩充节点内的信息表示。利用汉字与拼音的对应关系组织 Tire 树的节点，扩充传统 Trie 树节点内的信息，包含汉字、拼音、拆分结构，为后续的变体检索提供数据支持；

（2）增加节点间的联系。吸收 AC 自动机的失败指针机制，利用敏感词的重复子串，提高检索效率；

（3）融合规则的深度优先检索。依据中文变体的生成规则制定不同的检索规则，使算法最终能检索到多种敏感词变体。

3.1.1　扩充 Trie 树节点内的信息表示

中文 Trie 树的构建通常以单个汉字为节点建立 Trie 树，按常规构建方式，单个节点只存放单个汉字，本文为了支持后续的敏感词变体检索过程，扩充了节点内的信息表示，每个节点不仅存储该汉字，还存储其拼音、字形拆分结构，字形拆解通过汉字拆字字典获取[16]。Trie 树如图 3 所示，在词语结尾节点进行标注，例如：“他妈的”中“的”节点作为敏感词的最后一个字，插入了词语结束标志。此外，根据《通用规范汉字表》的分类，常用汉字共3 500个，若按常规方式构建 Trie 树，将导致其第 2 层节点过多，影响查询效率[17]。本文在为敏感词库构建 Trie 树时，第 2 层不直接存储敏感词首字，而是存储敏感词首字的拼音首字母，使拼音首字母相同的敏感词聚集在一个分支下，在检索时能根据敏感词首字的拼音快速定位到一个较小的分支下，加快检索速度[18]。

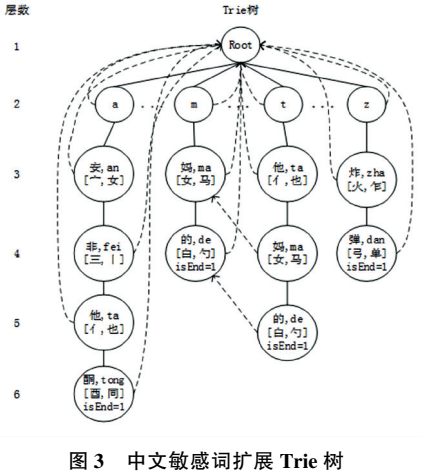

图 3　中文敏感词扩展 Trie 树

Fig. 3　Chinese sensitive words extended Trie tree

3.1.2　增加节点间的联系

AC 自动机的失败指针使其在检索失败时可以跳转到失败指针指向的节点，利用了模式串的相同子串，基于中文敏感词也存在相同子串的特点，本文为扩展 Trie 树添加失败指针，增加了节点间的联系。图 3 中的虚线指针即为失败指针，其优点是连贯了整个状态转移的过程，并且当存在相同子串时，可以从失败指针指向的节点继续检索，避免重新从根节点开始检索，提高了检索效率。构建节点间的失败指针的过程与传统 AC 自动机一致，但有两点

需要注意：

(1)构建失败指针时，节点之间仅通过节点汉字比较，与节点内汉字的拼音和字形拆解无关；

(2)因为第 2 层是字母索引，不直接存储敏感词信息，所以在寻找根节点的子节点时需要到第 3 层中寻找。例如，图 3 中“他妈的”的“妈”字节点构建失败指针时，先经其父节点的失败指针到达根节点，然后寻找根节点下的子节点“妈”，最终将失败指针指向“妈的”中的“妈”节点。

本文基于敏感词库构建的敏感词扩展 Trie 树的最大深度不超过 10，树中除第 1 层(根节点)，第 2 层(首字母索引)外，其余节点均存储汉字及其拼音和拆分信息以实现后续的变体检索，同时在敏感词最后一个字的节点存储敏感词信息，再通过建立失败指针缩短具有相同子串的敏感词的检索时间，并序列化保存生成的 Trie 树，加快后续检索时的读取速度。

3.1.3 融合规则的深度优先检索

本文的敏感词 Trie 树节点嵌入了多种信息，与传统 AC 自动机在节点只存储单一信息不同，在检索过程中对一个输入字符，可能出现从一个节点可以跳转到多个子节点的情况，也就是说其是一个非确定性有限状态自动机(NFA)。根据自动机理论，一个 NFA 可以转换为一个等价确定有限状态自动机(DFA)，敏感词 Trie 树显然可以在将其转换为 DFA 后用传统的 AC 自动机检索。但考虑到实现时的存储开销和检索效率，本文不直接转化为 DFA，而是在 AC 自动机检索机制上进行修改，称为 RSWET (Retrieval of Sensitive Words based on Extended Trie tree)算法，以实现多类敏感词变体检索，其核心是融合规则的深度优先检索，基于 AC 自动机的检索模式，对所有可能路径进行深度优先搜索，“规则”是指检索时节点与节点之间的跳转规则，是算法能检索到敏感词变体的关键，具体检索规则如下：

1)特殊字符混入的识别规则

当从待检测字符串读取到特殊字符时，直接忽略，读取字符串下一个字符。例如：字符串“妈 & 的”忽略了“&”后，能匹配敏感词“妈的”。

2) 拼音替换的识别规则

当待检测字符串读取到英文字母时，需要考虑是否是拼音替换的情况。如图 4(a)所示，当前节点在“妈”节点(虚线标识)，待检测字符串读取到“d”时，就需要与当前节点的子节点的拼音信息进行比较，如果待检测字符串“d”开头的字母序列能与子节点的拼音一致，即为可能的路径，当前节点跳转至子节点。如图 4(b)所示，当前节点跳转至“的”节点。

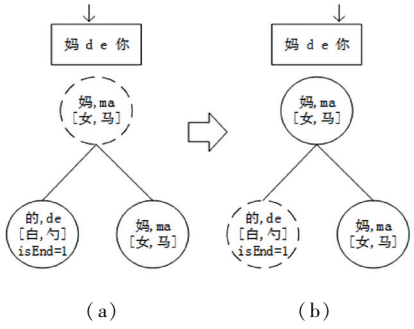

图 4 拼音替换的识别规则

Fig. 4 Recognition rules for pinyin substitution

3) 字形拆解替换的识别规则

当待检测字符串读取到汉字时，首先匹配汉字节点；若匹配不到，再考虑其是否为敏感词字形拆解后的组成部分，这一步需要用到拆分辅助表。拆分辅助表是一种哈希表，与节点绑定，存储该节点所有子节点汉字的拆分字信息。拆分字作为哈希表的键值(key)和值(value)是对应所有原字节点的指针集合。如图 5 所示，当前节点是“手”(虚线标识)，若输入拆分字为“木”或“仓”，则以其为 key 都可获得“枪”节点的指针，从而匹配“枪”字。同理，“王”、“不”可匹配“环”字。

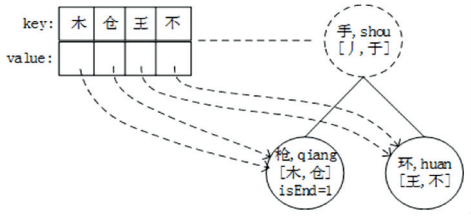

图 5 拆分辅助表

Fig. 5 Split auxiliary table

上述过程可看出，在对拆分汉字匹配时与拆分顺序无关，会导致误检问题。

4) 缩写的识别规则

为了能识别缩写形式，需要根据首字母或部分拆解信息进行匹配。例如，如图 6(a)所示，当前节点为 Root，待检测字符串读取到“s”，首先根据字母 s，定位到第 2 层的“s”节点，待检测字符串中以“s”开头的字母序列没有与“s”节点的任何子节点的拼音一致，进一步考虑拼音首字母为 s 的情况，“s”节点下的子节点首字母都是 s 开头的，均是可能的路径，所以图 6(b)中，当前节点有 3 个，“傻”，“睡”，

“手”，待检测字符串读取到“仓”，所有当前节点的拆解信息中，不含“仓”，并且“傻”节点和“睡”节点的子节点中也不存在“仓”，所以这两条路径就此终止，“手”节点的子字节“枪”的拆解信息中含有“仓”，即为可能的路径，图 6(c)当前节点转至“枪”节点。

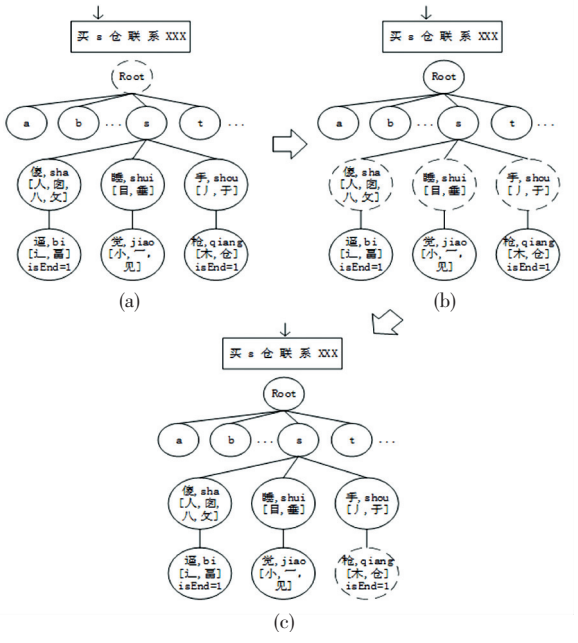

图 6　缩写的识别规则

Fig. 6　Recognition rules for abbreviations

当从当前节点无法匹配到子节点，即依据规则无法找到可行的路径时，当前节点会跳转至当前节点失败指针指向的节点。

针对 4 种变体的 4 种规则，使得 Trie 树具备识别变体的能力。

本文检索算法基于 AC 自动机和深度优先搜索的思路，吸收 AC 自动机通过失败指针加快具有相同子串的字符串匹配速度的优点，对整个 Trie 树进行深度优先搜索。为了提高检索效率，直接读取 Trie 树的序列化文件加载 Trie 树；Trie 树第二层使用拼音首字母作为索引，检索时能更快定位到一个较小的分支；Trie 树的父节点用字典存储子节点，以节点汉字作为索引 key 定位节点，辅助拆分表利用了哈希表结构定位节点，均避免了多次循环的时间消耗；因为中文敏感词的长度一般不超过十个字符，所以当存在多条路径时检索范围限制在后续的十个字符，起到剪枝的效果，能显著加快检索速度，且不会对检索结果产生影响。相较于其他基于规则的敏感词变体检测算法，本文提出的检索算法，只需要提供敏感词库，不需要再手动添加额外信息，即可检索变体，且能检测的变体范围更广，速度更快。

3.2　敏感词变体的二次判别

由于基于规则的检索且检索的变体范围广，不可避免的存在误检问题，会将正常语句中的部分字符串识别为敏感词变体，如图 7 所示，原文中的“我一”被识别为敏感词“我日”的变体形式，“我一”是“我日”经过拆解变成“我口一”再经过缩写，省略了“口”得到的变体，但是在原文里，“我一生中”是正常的语序，为了解决这个问题本文训练一个二分类器，对正常词和敏感词变体进行分类。

图 7　敏感词误检

Fig. 7　Misdetection of sensitive words

通过中文纠错数据集，将纠错前和纠错后的句子进行分词，提取出变更的短语[19]，如图 8 所示，纠错前的短语为负样本，纠错后的短语为正样本，总共提取出 60 万条短语，以此构成分类器的训练数据集。

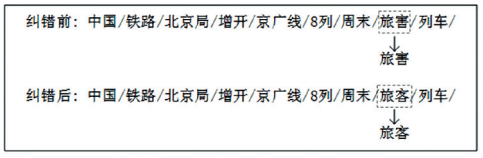

图 8　分词提取训练数据

Fig. 8　Segmentation extract training data

分类模型结合中文预训练模型 BERT 已经学习过大量中文文本序列的特点，在此基础上微调做分类任务[20]。BERT 的下游采用 BiLstm 模型，由正向 Lstm 和反向 Lstm 构成，能有效提取出文本序列特征，最后再使用全连接层输出分类结果。

3.3　实验分析

3.3.1　实验环境与评价指标

本实验环境 CPU 为 Intel i5-8250U，内存为 16 GB，操作系统为 Windows 10，编程环境 python 3.6。使用精准率(*Precision*)，召回率(*Recall*)和 *F*1 分数评价实验结果，见公式(1) ~ 公式(3)：

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{3}$$

其中，*TP* 是指实际为敏感词也被判定为敏感词的数目；*FP* 是指实际为正常词被判定为敏感词的数目；*FN* 是指实际为敏感词但被判定为正常词的数目。

3.3.2 数据集

实验数据集是从网上收集到 400 个敏感词变体，混入到 2 万字的正常文本中，组成测试文本，其中变形体涵盖了 4 种变体类型，部分敏感词变体举例见表 1。

表 1 敏感词变体举例

Table 1 Examples of sensitive word variants

敏感词原词	敏感词变体
侦听设备	亻贞口斤设备
破解	皮角
傻逼	sb
奸商	J 商

3.3.3 实验结果分析

将本文提出的 RSWET 算法与基于确定有穷自动机的 ST-DFA(Swift Tree DFA)算法，基于决策树的 RSWDT 算法做对比实验，构建 Trie 树使用的是同一开源敏感词库(不含敏感词变体)，共计2 500个敏感词，最终生成的 Trie 树大小见表 2。

表 2 不同算法 Tire 树大小

Table 2 Tire tree size for different algorithms

算法	Trie 树大小/KB
ST-DFA	316
RSWDT	433
RSWET	507

针对不同长度(20 字、200 字、2 000 字、20 000 字)的文本，各算法检索耗时(单位：字/ms)见表 3。可见本文提出的 RSWET 算法在耗时上具有一定优势，敏感词检索效率较高。

表 3 不同算法对不同长度文本检索耗时

Table 3 Different algorithms take milliseconds to retrieve text of different lengths ms

算法	20 字	200 字	2 000 字	20 000 字
ST-DFA	17.93	18.17	24.34	54.37
RSWDT	3.94	23.49	645.63	2 455.64
RSWET	0.22	5.69	14.14	88.00

为了进一步论证算法效率，对长文本进行测试，取十四万字的正常文本，在其中随机插入 20 个、200 个、2 000 个、20 000 个敏感词变体，其时间消耗见表 4。可见在十四万字文本中插入两万个敏感词后，其检索耗时约 2.9 s。与检索两千个敏感词相比，耗时显著增加的原因有两点：一是待检测文本的字符数明显增加，二是敏感词数量增多导致 Trie 树中节点匹配比较的次数增多，考虑到一般网页中并不会有这么多字符，RSWET 算法在网页敏感词检测中具有很强的实用性。

表 4 RSWET 检索长文本(十四万字)耗时

Table 4 RSWET retrieval of long text takes time ms

敏感词个数	耗时(毫秒/ms)
20	1 423.09
200	1 397.56
2 000	1 539.61
20 000	2 924.54

为了验证 BERT+BiLstm 分类模型的有效性，本文同时在中文纠错数据集上训练了 BERT+TextCNN 模型、单一 BERT 模型和 BERT+BiLstm 模型，实验结果见表 5，BERT+BiLstm 在准确率、召回率和 *F*1 分数上均优于另外两个模型。

表 5 不同分类模型结果比较

Table 5 Comparison of results between different classification models %

模型	*Precision*	*Recall*	*F*1
BERT+BiLstm	95.74	96.15	95.94
BERT+TextCNN	95.28	95.37	95.32
BERT	92.29	93.48	93.21

通过对敏感词变体数据集进行测试，不同变体检索算法结果见表 6。

表 6 不同变体检索算法结果比较

Table 6 Results of different variable search algorithms %

算法	*Precision*	*Recall*	*F*1
ST-DFA	97.00	40.50	57.14
RSWDT	93.26	65.75	77.13
RSWET	87.85	94.25	90.94
RSWET+分类器	98.69	94.25	96.41

由表 6 可知，本文提出的 RSWET 算法对比 ST-DFA 算法和 RSWDT 算法在召回率上有显著提升，达到了 94.25%，因为 ST-DFA 算法和 RSWDT 算法，RSWET 能识别到各种缩写变体，识别的变体类型更多，但是单一的 RSWET 精准率明显低于 ST-DFA 算法和 RSWDT 算法，误检情况更多，引入分类器后，对识别到的敏感词变体进行二次判别，显著减少了误检的情况，有效提升了精准率，*F*1 分数达到 96.41%。综上，RSWET+分类器的算法对敏感词变体能起到较好的识别效果。

4　结束语

针对网络中存在的中文敏感词变体问题，本文提出一种中文敏感词变体识别方案，该方案重点关注识别的敏感词变体种类数量以及识别的时间效率问题。总结了网络文本中出现的敏感词变体类型，根据中文敏感词变体特点构建出扩展 Trie 树，提出了融合规则的深度优先检索方法，能识别出多种中文敏感词变体类型，且通过基于 BERT 的二分类算法进行二次判别，有效降低了规则识别的误检率，提高了结果的精确率。实验结果表明，相比同类基于规则的变体识别方案能够识别的变体种类更广，精度更高，检索效率更高，能满足应用需求。但是敏感词变体形式多样，本文基于 Trie 树的检索无法覆盖全部变体类型，如同音字替换，相似字形替换等，有些词在不同语境下存在不同含义，需要结合语义进行识别，如何进一步扩大识别范围将是下一步工作的重点。

参考文献

[1] YOON T, PARK S Y, CHO H G. A smart filtering system for newly coined profanities by using approximate string alignment [C]// Proceedings of 2010 10th IEEE International Conference on Computer and Information Technology. IEEE, 2010: 643-650.

[2] 刘邦国，陈庆春，类先富. 一种面向 PDF 文本内容审查的高效多模式匹配算法[J]. 计算机应用研究，2020，37(6)：1755-1759.

[3] DEWANI A, MEMON M A, BHATTI S. Development of computational linguistic resources for automated detection of textual cyberbullying threats in Roman Urdu language [J]. 3 C TIC: Cuadernos De Desarrollo Aplicados A Las TIC, 2021, 10(2): 101-121.

[4] 李少卿，吴承荣，曾剑平，等. 不良文本变体关键词识别的词汇串相似度计算[J]. 计算机应用与软件，2015(3)：151-157.

[5] FU F, YU Y, WU X, A sensitive word detection method based on variants recognition [C]//Proceedings of 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). IEEE, 2019: 47-52.

[6] LIU G, FANG Y, LIU J Y. Sensitive Word-stock designing based on correlative word and extension rule [J]. Journal of Sichuan University (Natural Science Edition), 2009, 46(3): 667-671.

[7] 张会昌. 基于领域词典的中文文本相似度匹配[D]. 济南：山东大学，2014.

[8] 成彦衡，黄宇. 基于 K 近邻算法的网络敏感信息过滤方法[J]. 电子设计工程，2023，31(6)：105-108，113.

[9] 周昊，沈庆宏. 基于改进音形码的中文敏感词检测算法[J]. 南京大学学报(自然科学)，2020，56(2)：270-277.

[10] 王华敏，黄梦醒，冯文龙，等. 基于改进音形码与 HowNet 的中文词相似度检测算法[J]. 计算机仿真，2022，39(8)：460-465，472.

[11] 刘莹，杨超宇. 融合有向图的文本敏感词过滤模型[J]. 绥化学院学报，2022，42(2)：143-148.

[12] XUE P Q, NURBOL, WUXURISLA M. Sensitive information filtering algorithm based on text information network [J]. Computer Engineering and Design, 2016, 37(9): 2447-2452.

[13] 余敦辉，张笑笑，付聪，等. 基于决策树的敏感词变形体识别算法研究及应用[J]. 计算机应用研究，2020，37(5)：1395-1399，1405.

[14] 吴珊，李英祥，徐鸿雁，等. 基于改进的 Trie 树和 DFA 的敏感词过滤算法[J]. 计算机应用研究，2021，38(6)：1678-1682，1688.

[15] AHO A V, CORASICKM J. Efficient string matching: an aid to bibliographic search[J]. Communications of the ACM, 1975, 18(6): 333-340.

[16] Gitrlub Inc. Project webpage[EB/OL]. 2023-04-01. https://github.com/kfcd/chaizi.git

[17] 教育部国家语言文字工作委员会. 通用规范汉字表[M]. 北京：语文出版社，2013.

[18] 李江波，周强，陈祖舜. 汉语词典的快速查询算法研究[J]. 中文信息学报，2006(5)：31-39.

[19] WANG D, TAY Y, ZHONG L. Confusionset-guided pointer networks for Chinese spelling check[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 5780-5785.

[20] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv: 1810.04805, 2018.