基于上下文词预测和窗口压缩编码的数字水印方法

向凌云[1,2]，黄明豪[1]，张晨凌[1]，杨春芳[3]

（1. 长沙理工大学计算机与通信工程学院，湖南 长沙 410114；
2. 长沙理工大学综合交通运输大数据智能处理湖南省重点实验室，湖南 长沙 410114；
3. 信息工程大学河南省网络空间态势感知重点实验室，河南 郑州 450001）

摘 要：针对已有自然语言数字水印方法可替换词数量有限以及水印提取效率低的问题，提出了一种基于上下文词预测和窗口压缩编码的数字水印方法。该方法通过神经网络语言模型自动学习原始文本中每个词的上下文语义特征，预测每个词的候选词列表，从而扩充可用于嵌入水印信息的可替换词数量。同时，考虑到不同位置的候选词的替换对句子语义的影响存在差异，该方法以由多个词组成的窗口为单位来嵌入水印信息，并通过词替换前后句子间的相似度来优化水印嵌入时候选词的选择。在此基础上，提出了一种语义无关的窗口压缩编码方法，其根据窗口中词的字符信息对窗口进行水印编码，解决了提取水印信息时对词替换位置的原始上下文的依赖。实验结果表明，所提方法在具有较高嵌入容量和文本质量的前提下，大大提高了水印的提取效率。

关键词：数字水印；词替换；词预测；水印编码

中图分类号：TP309

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2024033

Digital watermarking method based on context word prediction and window compression coding

XIANG Lingyun[1,2], HUANG Minghao[1], ZHANG Chenling[1], YANG Chunfang[3]

1. School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
2. Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China
3. Henan Key Laboratory of Cyberspace Situation Awareness, Information Engineering University, Zhengzhou 450001, China

Abstract: To address the problems of limited number of substitutable words and low watermark extraction efficiency in the existing natural language digital watermarking methods, a creative method based on context word prediction and window compression coding was proposed. Firstly, the contextual semantic features of each word in the original text were automatically learned through a neural network language model, and then the candidate word set for each word was predicted, thus the number of substitutable words that could be utilized for carrying watermark information was expanded. Meanwhile, considering the difference of the semantic impact caused by the substitutions of candidate words at different positions, the watermark information was embedded into each window containing several words, and the selection of candidate words for watermark embedding was optimized by the similarity between sentences before and after performing word substitutions. Finally, a semantic-independent window compression coding method was proposed, which encoded each window as appointed watermark information in terms of the character information of words contained in the window. So that during watermark extraction, the dependence on the original context at the position of word substitution was eliminated. The experimental results show that the proposed method greatly improves the watermark extraction efficiency with high embedding capacity and text quality.

Keywords: digital watermarking, word substitution, word prediction, watermarking coding

收稿日期：2023−08−25；**修回日期**：2023−11−19

通信作者：杨春芳，chunfangyang@126.com

基金项目：国家自然科学基金资助项目（No.61972057, No.61872448）；湖南省自然科学基金资助项目（No.2022JJ30623）

Foundation Items: The National Natural Science Foundation of China (No.61972057, No.61872448), The Natural Science Foundation of Hunan Province (No.2022JJ30623)

0 引言

随着互联网和社交媒体的飞速发展，通过网络高效传输和便捷分享多媒体数据成为人们工作和生活中不可或缺的一部分。但多媒体数据的易于编辑特性也使它们容易被非授权使用、恶意传播或篡改等，从而引发侵权等安全问题[1]。为此，数字水印技术作为一种有效的手段广泛应用于多媒体版权保护，并成为信息安全领域的一个研究热点。

数字水印技术利用多媒体数据中存在的冗余空间来嵌入标识多媒体数据版权的水印信息，实现对多媒体数据版权的认证。数字水印技术已经在图像[2]、视频[3]和音频[4]等领域取得了丰富的成果，但在电子文档方面的应用还相对较少[5]。

目前，电子文档数字水印技术有2种主要类型：基于文档结构和基于文本内容。基于文档结构的方法使用字间距、行间距、字形[6]、字库[7]等格式和结构上的修改来嵌入水印信息。这些能够嵌入水印信息的空间且与文档具体内容无关，早期的方法难以抵抗重排版、光学字符识别（OCR, optical character recognition）的攻击，且多数方法效率比较低，难以抵抗针对具体内容的修改攻击[8]。基于文本内容的方法则使用字词或句法上的变换在文本内容中嵌入水印信息[9-10]。水印信息的嵌入不会破坏文本的整体语义，与基于文档结构的方法相比，具有强鲁棒性。

基于句法的数字水印方法主要是通过对整个句子的结构进行修改来实现信息的嵌入[11-12]。尽管不同的语种具有不同数量的可以进行语义等价的句法变换，但可用于水印嵌入的句法变换数量有限，实现难度大，且以句子为单位的水印嵌入使这类方法的水印容量极低。因此，已有的基于文本内容的水印研究集中在词级别的修改来实现水印信息的嵌入。这类方法利用文本中的各种词汇和语法特征，通过改变文本局部的词语（如利用同义词替换）来嵌入水印信息[13]。早期基于同义词替换的方法通过将含义相同的同义词编码成不同的值来表示不同的水印信息[14]，再通过同义词的替换来嵌入指定的水印信息。但直接的同义词替换容易出现语法和语义的错误而降低文本质量，因为一个词可能有多种词性和词义。因此，后续研究工作在同义词替换时考虑了词性的一致、搭配词的合适度等[15]，降低了同义词替换引起的语义失真，同时也进一步降低了可进行同义词替换的数量，导致文档的水印容量不高。

随着深度神经网络的兴起、大规模数据集的不断发展，神经网络语言模型在自然语言处理领域取得了重要的突破。研究者也开始将神经网络语言模型应用于数字水印领域，从同义词之间的替换扩展到合适词之间的替换，大大提高了水印容量。文献[16]利用预训练模型BERT（bidirectional encoder representation from transformers）对句子进行掩码预测，为句子中固定位置生成候选词，通过对候选词进行编码，在水印信息的约束下选择对应的候选词来替换原词。为了优化性能，在文献[16]的基础上，文献[17]根据嵌入前后句子的语义相似度（SS, semantic similarity），对候选词进行了过滤。与已有基于同义词替换的方法相比，尽管这2种方法提高了水印容量和水印文本质量，但由于神经网络语言模型的使用，水印嵌入和提取均需要耗费较多的计算资源，效率低。此外，这2种方法在水印提取时，均需要预测出与水印嵌入时相同的候选词列表，以正确解码选定候选词所嵌入的水印信息。因此，文献[16]采取固定位置掩码，确保掩码位置的上下文不被修改，但固定的位置以及候选词的简单筛选容易生成低质量的水印文本。文献[17]则设计了严格的替换条件，要求词替换前后生成一致的候选词列表，且替换后不能影响前一个已嵌入水印信息位置的候选词列表。严格的替换条件保证了水印信息的正确嵌入和提取，但也容易导致符合嵌入要求的单词过少，使嵌入容量过低。针对上述基于文本内容的数字水印方法存在的问题，本文提出了一种基于上下文词预测和窗口压缩编码的数字水印方法，以解决基于句法变换和同义词替换的方法的水印容量偏低，以及基于神经网络语言模型的词替换水印方法的效率低的问题。

已有方法在提取水印时，通常需要水印嵌入方共享大量的参数或模型，如同义词库、句法分析器、预训练掩码模型等，以保证水印的嵌入和提取采用的编码结果一致，但影响了水印提取的效率和安全性。尤其是基于神经网络语言模型的词替换水印方法，其提取效率非常低。因此，本文提出了一种窗口压缩编码方法，使水印提取时不需要其他辅助的信息或模型，仅依靠水印文本所包含的词语即可正

确提取水印信息，消除了对水印嵌入位置的候选词列表的依赖，大大提高了水印提取的效率。由于窗口压缩编码的编码结果具有不确定性，而每一种水印信息均需要对应一种文本内容。因此，为了保证每一种水印信息值均可以映射到一种可替换的文本内容，即水印信息值等于该文本内容经压缩后的编码值，可相互替换的文本内容的数量应尽量多，且数量越多，水印容量将越大。为了解决该问题，本文利用深度神经语言模型为文本中的词自动预测生成可替换词列表，突破同义词数量的限制。其次，以多个词组成的窗口为单位来进行压缩编码，从而增加可替换的窗口数量，并保证可替换窗口能够编码任意的水印信息。可替换的窗口数量增多，以及压缩编码的不确定，可能导致多个窗口编码成相同的值，因此，本文考虑使用预训练语言模型 RoBERTa 生成原始和替换后句子的向量表示，通过计算向量间的距离度量替换前后句子之间的语义相似度。在水印信息值的约束下，优先选择语义相似度高的候选句。

根据上述分析，本文工作的主要贡献如下。

1) 采用深度神经网络 Transformer 构建词预测模型，为文本中的词预测可替换的候选词列表。相比于已有的词替换水印方法，能够获得更多符合上下文语义的可替换词，并通过替换前后句子的语义相似度优化可替换候选词的选择，以提高替换后句子的质量。

2) 提出了一种窗口压缩编码方法来实现水印信息的高效提取。该方法以窗口中内容为编码对象，将编码过程和词预测过程分离，使提取时不需要再次通过词预测获取与水印嵌入时相同的候选词列表即可提取出水印信息，提高了水印提取效率。

3) 从水印容量、水印质量、水印提取效率等方面出发设置了丰富的对比实验，实验结果验证了本文方法的高性能，特别是水印提取效率上大大超过了已有基于词替换的自然语言水印方法。

1　相关工作

本文提出的数字水印方法本质上是一种基于词替换的自然语言信息隐藏方法，这类方法的早期研究成果主要集中在利用语义相同或近似的同义词的相互替换来实现水印信息或秘密信息的嵌入。理论上，这些同义词的替换能保证文本内容的语义不变，使嵌入的水印信息不容易被察觉，且不影响原始文本的使用价值，但能够提供用于版权认证的水印信息。Winstein[14]提出了第一个基于同义词替换的信息隐藏系统，该系统使用在任何情况下语义都等价的绝对同义词进行编码，并实现语义等价替换。绝对同义词数量极其有限，导致嵌入容量低。因此，Bolshakov 等[15]在嵌入过程中使用预先验证且语义兼容的相对同义词进行替换。但替换后的水印文本中容易存在词搭配错误，语义损失较大[18]，隐蔽性有待进一步增强。Yang 等[19]则通过构建形容词和副词的同义词库，并利用扩频技术来嵌入信息，降低词替换对原始文本的影响，以提高嵌入信息的隐蔽性。为了提高基于中文同义词替换的水印方法的鲁棒性，林建滨等[20]利用自动消歧技术，设计词汇和义项相似度的度量方式，根据词汇和义项相似度的值来进行同义词的选择与替换，从而提高水印文本抵抗机器消歧攻击的能力。

受限于同义词的数量，基于同义词替换的信息隐藏方法的嵌入容量普遍不高，极大地影响了这类方法的实用性。为了提高嵌入容量，文献[16]利用掩码语言模型来生成词的可替换词列表，取代了从同义词库中识别可替换的词，并进一步扩充了词替换的范围。由于词预测对上下文的依赖，为解决水印信息正确提取的问题，对嵌入位置进行了固定。文献[17]使用 BERT 模型生成候选词，并根据嵌入前后句子的语义相似度对候选词进行过滤，以提高信息嵌入后文本的质量。但为了保证替换前后的语义不变，设计了严格的候选词筛选规则，导致嵌入容量低。Zheng 等[21]则在文献[16]的基础上放宽了候选词的限制，虽然有效提升了嵌入容量，但容易改变原语句的语义信息。因此，为了适应对文本内容失真要求高的场景，在经过词替换后，文献[22-23]设计了可逆的自然语言水印嵌入方案，在水印提取的同时能够恢复被替换的原词。

除了替换词的范围和选择策略会影响自然语言信息隐藏方法的嵌入容量和词替换后文本的质量，替换词的编码方式也是影响信息隐藏方法性能的重要因素。杨潇等[24]将同义词数值化后利用矩阵编码，减少对文本内容进行修改操作的数量，同时嵌入尽可能多的信息。Xiang 等[25]则利用串表压缩算法 LZW（Lempel-Ziv-Welch）对秘密信息进行压缩，提取后再进行解压缩，在同样的载体文本中能够嵌入更多的信息。Yang 等[26]使用长短期记忆

（LSTM, long short-term memory）网络自动学习文本中的统计特征来对下一个生成位置进行预测，根据候选词的预测概率的分布对候选词进行定长编码或哈夫曼编码，嵌入容量和嵌入信息后文本质量都达到了不错的性能。为了避免对候选词采用定长编码时选择了不适合的词来嵌入信息，文献[27]通过集成多个位置上词的选择来优化单个位置上词的选择，以此来提高文本质量。

尽管不同的研究人员对基于词替换的信息隐藏方法进行了不同的优化，但水印嵌入和水印提取过程中大都采用了一致的方式获取相同的可替换词列表，以正确编码和解码所嵌入的水印信息，即水印嵌入和水印提取需要保持同步的可替换词列表和编码结果。这通常需要在水印嵌入和水印提取端共享大量的数据、参数或模型，如同义词库或预训练语言模型，从而导致水印提取的效率低。因此，本文提出的基于词替换的数字水印方法将重点解决水印提取效率低的问题。

2 方法描述

为了扩大可替换词的数量以及消除水印提取对失真上下文的依赖，提高水印提取效率，本文基于上下文预测和窗口压缩编码，提出了一种新的基于词替换的数字水印方法。为了获得更多高质量的候选词，构建了候选词预测器，利用原词丰富的上下文信息来预测适合当前上下文并替换原词的候选词列表。在通过词替换嵌入水印信息后，水印文本中相同位置的上下文可能发生改变，造成上下文的失真，从而使水印嵌入和提取过程中相同的位置，可能预测到不一样的候选词列表。

已有的水印方法通常利用某种规则对候选词列表中的候选词进行排序，再对候选词进行编码，候选词的编码值与其在排序后的候选词列表中位置有关。水印嵌入时选择编码值与水印值一致的候选词替换原词（当候选词与原词相同时，则保持不变）；水印提取时则通过相同的方式对候选词进行编码，获得含水印的词的编码值，即所嵌入的水印信息。为了准确提取水印信息，必须保证相同位置上的词在原始文本和水印文本中具有相同的编码值。若候选词列表发生改变，则容易导致相同的词在嵌入前后的编码值发生改变，水印信息提取失败。为了解决该问题，本文设计了一种独立于候选词列表的水印编码方式。该编码方式以多个词组成的窗口为单位来进行水印编码，且仅依赖窗口中词本身来进行压缩编码，简化编码过程，提高水印提取效率。

2.1 总框架

本文提出的数字水印方法主要分为 2 个过程，水印嵌入过程和水印提取过程。嵌入过程的整体框架如图 1 所示，主要涉及 3 个模块：基于上下文的候选词预测、窗口压缩编码以及水印嵌入优化。

1) 基于上下文的候选词预测

设置滑动窗口，依次对滑动窗口中的词进行掩码，再通过大规模预训练语言模型 Transformer[28] 进行二次训练，强化学习特定领域文本的语义特征，根据掩码位置的上下文实现对掩码位置的词预测，生成可替换掩码位置原词的候选词列表。将所获得候选词列表中的词依次替换相应原词后可得到候选窗口。

2) 窗口压缩编码

对于所有的候选窗口，根据窗口中词的字符编码信息设计压缩编码规则，得到候选窗口的水印编码值。对于不同语言的文本内容，通过相应的字符编码方式，如英文字母采用 ASCII 编码、中文字符采用 UNICODE 编码等，获得窗口中词的字符编码信息，再经过异或和模运算进行压缩获得窗口的水印编码值。

3) 水印嵌入优化

在水印信息的约束下，选择水印编码值与待嵌入水印值一致的候选窗口。由于压缩编码结果的不确定性，可能存在多个候选窗口编码值一致的情况。因此，通过计算候选窗口替换原始窗口前后句子的语义相似度，选择语义相似度最高的候选窗口来替换原始窗口实现水印信息的嵌入，从而优化词替换的位置，降低句子语义的失真。

与已有词替换的水印方法相比，本文方法的水印提取过程简单，效率高，仅需要对水印文本设置相同的滑动窗口，对窗口内容进行压缩编码，确定窗口的编码值，即可正确提取出水印信息。

2.2 基于上下文的候选词预测

对于通过词替换来不可感知地修改文本内容实现水印信息嵌入的方法，获取数量多、质量高的候选可替换词对提高水印文本质量以及水印容量至关重要。可替换词的获取可视为一个原词所在位置的词预测任务。词预测任务是自然语言处理领域的重要下游任务之一，已有较多的相关研究成果。

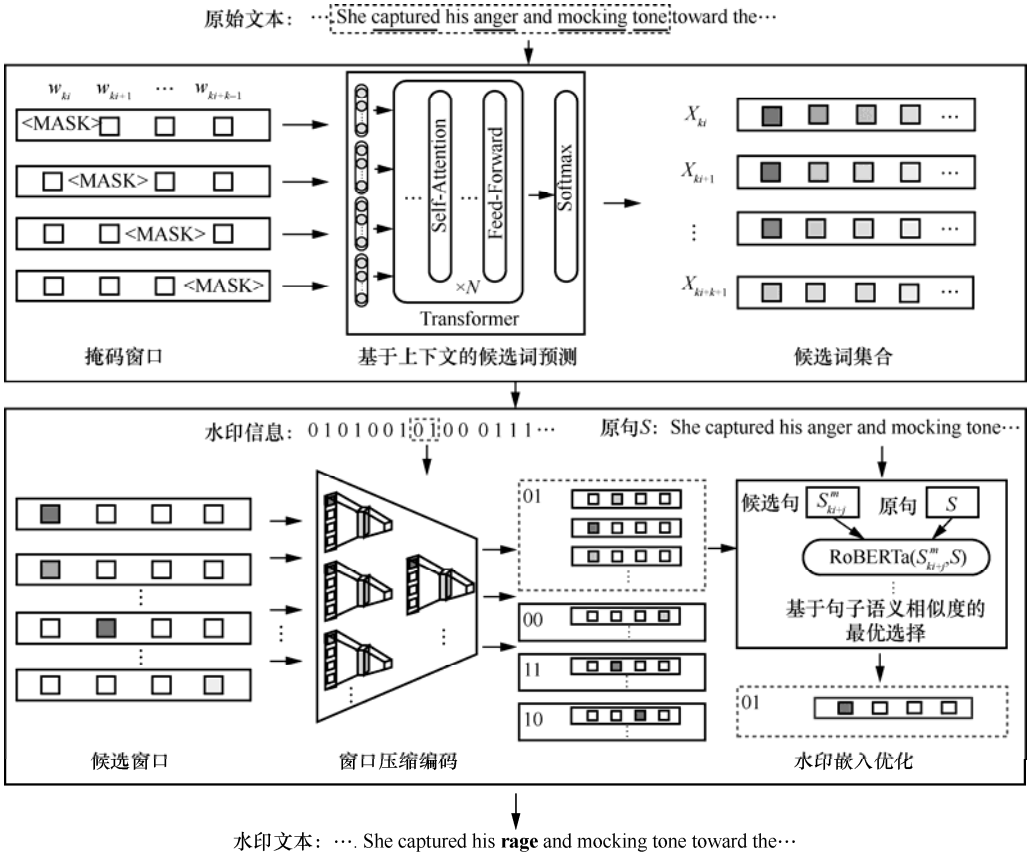


图 1　嵌入过程的整体框架

文献[29]基于共现概率和 N-grams 模型来确定最有可能成为候选词的单词列表，但是在预测过程中忽略了句子的语法和语义信息。文献[30]提出了一个基于生成对抗网络（GAN, generative adversarial network）的网络模型，以产生更真实的条件和非条件文本样本。文献[31]使用 Transformer 作为文本填充的词预测模型，通过自注意力机制来捕捉上下文关系，这种机制使 Transformer 模型可以在更长的序列中捕捉到更多的上下文信息，从而得到更合适的候选词，并通过实验验证了该模型的先进性。因此，本文使用以多头自注意力机制为基础结构的 Transformer 模型作为候选词预测器，为原始文本中的词生成可替换的候选词列表。

为了提高水印嵌入成功率，本文方法以多个词组成的窗口为单位进行水印信息的嵌入，因此，从文本的第一个词开始，每 k 个词划分为一个窗口，由此将原始文本划分成多个独立的窗口，通过对窗口中词进行特定的替换来嵌入水印信息。考虑到文本中存在一些经常出现但对文本的语义信息贡献较少的停用词，如常见的介词、连词、冠词等，适合替换这些词的候选词通常较少，且这些词所在位置不适合用于嵌入水印信息，因此，在具体划分窗口时，将根据事先整理好的停用词表对文本中词进行过滤，跳过停用词和标点符号而只对剩余的词进行窗口的划分。停用词和标点符号将不参与窗口中词的计数、词预测和窗口编码。

设待嵌入水印信息的原始文本为 $T=\{w_0,w_1,w_2,\cdots,w_{n-1}\}$，其中，$w_i$ 表示文本中第 i 个词，n 表示文本中词（不含停用词）的总数。第 i 个窗口为 $W_i=\{w_{ki},w_{ki+1},\cdots,w_{ki+k-1}\}$，$i=0,1,\cdots,\left\lfloor\frac{n}{k}\right\rfloor$，从窗口中第一个词开始依次进行掩码，即依次替换为屏蔽字符<MASK>，再通过所在句子的上下文语义特征的学习，来预测被掩码位置的候选词列表。设窗口中第 j+1 次被掩码的词为 w_{ki+j}，得到掩码句子 $S_{ki+j}=\{w_z,w_{z+1},\cdots,<\text{MASK}>,\cdots,w_{z+q-1}\}$，其中，词 w_z 是句子 S_{ki+j} 中的第一个词，q 是该句子中词的个数。将 S_{ki+j} 输入已训练好的 Transformer 模型中，可获得掩码位置词 w_{ki+j} 的候选可替换词列表。

将掩码句子 S_{ki+j} 输入 Transformer 模型时，首先进行句子的序列化，即进行词嵌入处理和位置编

码处理。词嵌入处理是将句子中的词转换成词嵌入向量的形式，挖掘词与词之间的关系，使意思相近的词有相似的词嵌入向量。常见的获取词嵌入向量的方式如下：使用预训练模型直接生成；在语言模型训练过程中作为一个可训练的参数参与梯度优化，在模型训练完成时生成词嵌入向量。本文使用后者来获取文本中每个词的词嵌入向量，即在语言模型训练时保存词汇表中每个词所占的权重作为每个词的词嵌入向量表示。

对于掩码句子 S_{ki+j}，将其所包含的词在词汇表中的编号输入词嵌入处理函数中，通过训练得到每个词的词嵌入向量表示，再获得整个掩码句子的嵌入矩阵，具体计算过程为

$$\mathbf{WE}(S_{ki+j})=\text{Embedding}(\{V(w_{z+r})\},d_{\text{model}})\text{sqrt}(d_{\text{model}}) \tag{1}$$

其中，$V(w_{z+r})$ 表示句子中第 r 个词 w_{z+r} 在词汇表中的编号，$\{V(w_{z+r})\}$ 表示句子中所有词在词汇表中的编号集合，d_{model} 表示转换的向量维度，sqrt(·) 表示算数平方根函数，Embedding(·) 表示词嵌入处理函数，**WE** 表示词嵌入矩阵。

根据式(2)和式(3)依次计算句子中每个词的位置编码，获得位置编码矩阵 **PE**

$$\mathbf{PE}(\text{pos}(w_{z+r}),2\text{index})=\sin\left(\frac{\text{pos}(w_{z+r})}{10\,000^{\frac{2\text{index}}{d_{\text{model}}}}}\right) \tag{2}$$

$$\mathbf{PE}(\text{pos}(w_{z+r}),2\text{index}+1)=\cos\left(\frac{\text{pos}(w_{z+r})}{10\,000^{\frac{2\text{index}}{d_{\text{model}}}}}\right) \tag{3}$$

其中，$\text{pos}(w_{z+r})$ 函数返回词 w_{z+r} 在句子 S_{ki+j} 中重新编号后的索引，取值范围为 $0,1,\cdots,q-1$；index 为嵌入维度，取值范围为 $1,2,\cdots,d_{\text{model}}$。

将词嵌入矩阵和位置编码矩阵相加，得到掩码句子的位置嵌入矩阵 **IE**

$$\mathbf{IE}(S_{ki+j})=\mathbf{WE}(S_{ki+j})+\mathbf{PE}(S_{ki+j}) \tag{4}$$

将 **IE** 作为 Transformer 模型第一个编码器的输入，剩余编码器的输入为上一个编码器的输出。编码器采用多头注意力机制进行数据处理，可表示为

$$\text{head}_u=\text{Attention}(\boldsymbol{QW}_u^{\boldsymbol{Q}},\boldsymbol{KW}_u^{\boldsymbol{K}},\boldsymbol{VW}_u^{\boldsymbol{V}}) \tag{5}$$

$$\text{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{Concat}(\text{head}_1,\cdots,\text{head}_h)\boldsymbol{W}^{\boldsymbol{O}} \tag{6}$$

其中，$\boldsymbol{W}_u^{\boldsymbol{Q}}$、$\boldsymbol{W}_u^{\boldsymbol{K}}$和$\boldsymbol{W}_u^{\boldsymbol{V}}$ 分别为查询矩阵 $\boldsymbol{Q}$（Query）、键值矩阵 $\boldsymbol{K}$（Key）和内容矩阵$\boldsymbol{V}$（Value）所对应的随机初始化矩阵，$\boldsymbol{W}^{\boldsymbol{O}}$ 为权重矩阵。

具体来讲，对于第一个编码器，首先对位置嵌入矩阵 $\mathbf{IE}\in\mathbb{R}^{q\times d_{\text{model}}}$ 进行划分，平均分配到各个 head 中并行计算，在每个 head 中，**IE** 分别与 $\boldsymbol{W}_u^{\boldsymbol{Q}}$、$\boldsymbol{W}_u^{\boldsymbol{K}}$和$\boldsymbol{W}_u^{\boldsymbol{V}}$ 相乘得到对应的查询矩阵 $\boldsymbol{Q}$、键值矩阵 $\boldsymbol{K}$ 和内容矩阵 $\boldsymbol{V}$。计算查询矩阵 $\boldsymbol{Q}$ 和键值矩阵 $\boldsymbol{K}^{\text{T}}$ 的乘积并除以 $\sqrt{q}$，使用归一化指数函数 Softmax(·) 计算每一个词对于其他词的 Attention 系数，将系数矩阵和内容矩阵 $\boldsymbol{V}$ 相乘得到当前 head 的输出，MutilHead(·) 函数将所有 head 的输出进行拼接后乘以权重矩阵 $\boldsymbol{W}^{\boldsymbol{O}}$，得到下一编码器的输入 $\mathbf{IE}'\in\mathbb{R}^{q\times d_{\text{model}}}$。

通过最后一个编码器的输出将得到当前句子 S_{ki+j} 中每个词的上下文向量表示，记为 $\boldsymbol{C}_{ki+j}=\{\boldsymbol{c}_0,\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_{q-1}\}$，其中，$\boldsymbol{c}_g$ 表示句子中第 g 个词的上下文向量。

将当前掩码位置的上下文向量 $\boldsymbol{c}_b$ 投影到词汇表中，将得到词汇表中词的预测得分的分布为

$$P(v_r\,|\,\boldsymbol{C}_{ki+j},b)=\text{Softmax}(\boldsymbol{V}\boldsymbol{c}_b) \tag{7}$$

其中，$\boldsymbol{V}$ 为词汇表向量矩阵，v_r 为矩阵中的第 r 行所对应的词。

设置候选词阈值 α 筛选出当前掩码位置，即被掩码词 w_{ki+j} 的候选可替换词。当 $P(v_r\,|\,\boldsymbol{C}_{ki+j},b)\geqslant\alpha$ 时，将 v_r 所表示的词添加到候选词列表 X_{ki+j} 中。X_{ki+j} 中的词为初筛选后较适合替换文本中第 $ki+j$ 个词 w_{ki+j} 的候选可替换词。

通过上述相同的操作，将依次得到当前窗口 $W_i=\{w_{ki},w_{ki+1},\cdots,w_{ki+k-1}\}$ 中每个词的候选词列表，分别记为 $X_{ki},X_{ki+1},\cdots,X_{ki+k-1}$。

2.3 窗口压缩编码

已有水印方法通常直接对整个可替换词列表进行编码，根据某种确定的方式对列表中词进行排序，每个词将编码成唯一的值。因此，在水印提取时除了水印文本内容以外，还需要依赖整个可替换词列表，即保证水印嵌入和水印提取具有相同的可替换词列表以正确提取水印信息。但当进行词替换以嵌入水印信息时，将导致后续词的上下文发生改变，使替换前后所预测的后续词的可替换词列表存在差异，影响后续水印信息的正确提取。因此，本文提

出了一种新的窗口压缩编码方法，对每个用候选词替换原词所得的候选窗口以及原窗口独立进行编码，不依赖于其他候选窗口进行水印的编码和解码，从而仅依靠水印文本中窗口内容即可完成水印的提取，避免了提取时需要生成候选可替换词列表等一系列复杂的过程，大大提高了提取效率。

窗口压缩编码方法的基本思想是利用窗口中所包含词的字符编码值经过一系列的异或和模运算将窗口编码成固定长度的水印编码值。针对不同语种的文本内容，采用相应的字符编码方式对窗口中所包含词进行编码。由于窗口中词通常会编码成多个字节，因此，首先将窗口中每个词的字符编码结果按照字节进行异或，再进行模运算将每个词编码成固定比特长度的值，最后将窗口中所有词的编码值进行异或，得到窗口的水印编码值。

对于窗口 $W_i=\{w_{ki},w_{ki+1},\cdots,w_{ki+k-1}\}$，将其压缩成固定长度为 d bit 的水印编码值 E_i 的具体计算过程为

$$l_{ki+j}=\left(\mathrm{ef}\left(w_{ki+j}^{1}\right)\oplus\mathrm{ef}\left(w_{ki+j}^{2}\right)\oplus\cdots\right)\bmod(2^d) \tag{8}$$

$$E_i=l_{ki}\oplus l_{ki+1}\oplus\cdots\oplus l_{ki+k-1} \tag{9}$$

其中，l_{ki+j} 为词 w_{ki+j} 的编码值；$\mathrm{ef}(w_{ki+j}^{1})$ 为词 w_{ki+j} 的字符编码结果的第 1 个字节，如果 w_{ki+j} 是英文单词，则 $\mathrm{ef}(w_{ki+j}^{1})$ 表示该单词的第 1 个字母的 ASCII 码值，如果 w_{ki+j} 是中文词语，则 $\mathrm{ef}(w_{ki+j}^{1})$ 表示该词语的 UNICODE 编码结果的第 1 个字节；$\oplus$ 为异或操作，对异或结果进行模运算，使词 w_{ki+j} 的编码值 l_{ki+j} 长度为 d。

2.4　水印嵌入优化

水印嵌入时，设当前窗口待嵌入的长度为 d 的水印信息为 M_i。若当前窗口的水印编码值 E_i 与 M_i 一致，则不进行替换，滑动窗口至下一窗口继续进行下一个 d bit 水印信息的嵌入；否则，通过候选词替换窗口中的原词来调整窗口的水印编码值为 M_i，实现水印信息的嵌入。

窗口中每个词均有相应的候选词列表，候选词列表中包括了多个可以用来替换原词的候选词。为了选择合适的候选词进行替换以提高水印文本的质量，本文利用句子的语义相似度，对水印嵌入过程进行了优化。首先，将当前窗口内的每个词依次用其对应的候选词列表中的词进行替换，得到多个候选窗口，并利用窗口压缩编码规则计算每个候选窗口的水印编码值。然后，保留编码值与 M_i 一致的候选窗口所对应的候选词到列表 Y_i 中。由于 Y_i 中的候选词可能有多个，因此，将根据语义相似度选择最优的候选词来替换原词，从而优化水印嵌入过程。具体将计算 Y_i 中每个候选词替换原词后所得句子与原句的语义相似度，选择语义相似度最高的候选词进行词替换以使句子语义失真最小。

为了衡量 2 个句子之间的相似度，本文采用预训练模型 RoBERTa[32]来获取句子嵌入向量，再通过计算 2 个句子嵌入向量的余弦距离来度量候选句（将原词替换为候选词后的句子）和原句的语义相似度，完成对候选词的选择。在候选词的预测过程中，本文使用的 Transformer 模型更加关注候选词是否符合上下文语义，而 RoBERTa 提高了词汇表的丰富度和表征能力，能够更好地对文本进行编码，更准确地度量候选句与原句的语义相似度。

设水印编码值与水印信息 M_i 一致的候选窗口所对应的候选词列表为 $Y_i=\{y_{ki},y_{ki+1},\cdots,y_{ki+k-1}\}$，其中，$y_{ki+j}$ 表示第 i 个窗口中的第 j 个词 w_{ki+j} 对应的符合编码条件的候选词列表。将 y_{ki+j} 中的第 m 个候选词替换原始文本中的原词 w_{ki+j} 得到修改后的候选句，记为 S_{ki+j}^{m}。将 S_{ki+j}^{m} 和原句 S 输入已预训练好的 RoBERTa 模型中进行自动学习，获得 2 个句子的句子嵌入向量后，计算句子嵌入向量之间的余弦相似度来度量候选句和原句之间在语义上的相似程度。候选句 S_{ki+j}^{m} 和原句 S 的语义相似度计算式为

$$\mathrm{Sim}_{ki+j}^{m}=\mathrm{RoBERTa}(S_{ki+j}^{m},S) \tag{10}$$

依次计算 Y_i 中所有候选词所在候选句与原句的语义相似度后，求相似度最高时的 m 值和 j 值，确定相似度最高的候选词所在位置，计算式为

$$(s,t)=\arg\max_{m,\,j}\mathrm{Sim}(S_{ki+j}^{m}) \tag{11}$$

此时从候选词列表 Y_i 中选择语义相似度最高的词为 y_{ki+t}^{s}，然后利用 y_{ki+t}^{s} 将原窗口中的第 t 个词 w_{ki+t} 替换，得到含水印的窗口，实现水印信息的优化嵌入。

2.5　基于压缩编码的水印提取

本文方法从所生成的水印文本中提取水印信息时，不需要进行复杂的可替换候选词生成，只

需采用与嵌入过程中一致参数的窗口压缩编码方法即可提取出所嵌入的水印信息，大大降低了水印提取算法的时间和空间复杂度，提高了水印提取效率。

给定水印文本 $T'=\{w'_0,w'_1,w'_2,\cdots,w'_{n-1}\}$，使用与嵌入水印时一致的窗口大小 k（即窗口中词的个数）和窗口编码长度 d 来提取水印信息。首先，利用相同的停用词表和标点符号集合过滤水印文本中的词，将剩余词每 k 个划分为一个独立的窗口。设第 i 个窗口的内容为 $W'_i=\{w'_{ki},w'_{ki+1},\cdots,w'_{ki+k-1}\}$，利用 2.3 节所提出的窗口压缩编码方法计算当前窗口 W'_i 的水印编码值，即利用式(8)根据窗口中每个词 w'_{ki+j} 的字符编码信息，得到其长度为 d 的编码值 l'_{ki+j}；再利用式(9)将窗口内所有词的编码值进行异或得到当前窗口的水印编码值 E'_i，即窗口 W'_i 中所嵌入的水印信息。所有窗口的水印编码值串联起来将得到嵌入的整个水印信息。

3 实验与结果分析

3.1 实验设置

本文选取互联网上最常见的英文文本数据集 News 来训练候选词预测模型，其包含来自纽约时报、Breitbart、CNN 等 15 种刊物的文章，约 196 万条句子。在模型训练之前，对 News 数据集进行预处理，主要包括将所有字母小写化，删除特殊符号、表情、网页链接等。

使用 PyTorch 1.6.0 仿真平台，基于 NVIDIA GeForce GTX 3070 GPU 和 CUDA11.0 加速模型训练。对于候选词预测模型，学习率初始设为 0.000 1，编码器层数为 6，每层 head 数为 8，dropout 率为 0.3。

使用困惑度（PPL, perplexity）评估生成的水印文本的质量。PPL 是一种常用的有参考文本质量评价指标，常用于评估语言模型的好坏。PPL 越小，表示模型的预测结果越准确，生成的文本质量越好。同时，本文使用预训练语言模型 stsb-roberta-base-v2 计算水印句与原句的句子嵌入向量之间的余弦距离，从而度量水印句与原句的语义相似度。

3.2 不同参数对文本质量的影响分析

本文测试了不同窗口大小 k、候选词阈值 α 和窗口编码长度 d（即窗口中嵌入水印信息的比特长度）对水印文本的影响。

窗口大小 k 的值越小，则候选窗口数将越少，而候选窗口的水印编码值具有一定随机性，容易导致没有候选窗口的水印编码值能够满足嵌入指定水印信息的要求，从而导致水印信息的嵌入失败。本文设置 $\alpha=0.02$，k=4,5,6，d=1,2 bit 生成水印文本，并对水印文本的质量进行了评估，实验结果如表 1 所示。

表 1 不同窗口大小和编码长度下的水印文本质量

k	PPL		SS	
	$d=1$ bit	$d=2$ bit	$d=1$ bit	$d=2$ bit
4	65	80	92.09%	87.98%
5	64	79	92.99%	88.91%
6	**62**	**72**	**93.88%**	**90.43%**

从表 1 可以发现，随着窗口中词个数 k 的增加，不论是在窗口中嵌入 1 bit 还是 2 bit 水印信息，水印文本的 PPL 都随之降低，PPL 越低，说明水印文本的质量越高；水印文本和原始文本之间的 SS 也呈现递增的趋势，说明窗口越大，所生成的水印文本能够越好地保持原始文本语义不变。这是因为窗口越大，可供选择的候选窗口数量越多，能够从中选择更优的窗口。当 k=6 时，水印文本与原始文本的语义相似度最高，且在每个窗口嵌入 2 bit 水印信息的情况下，相似度仍达到了 90.43%，说明在嵌入更多水印信息的时候，水印文本仍能较好地保持原始文本的语义。

窗口大小和窗口中嵌入水印的长度将影响候选窗口的数量和选择，进而影响水印文本的质量；同时，窗口中每个词的可替换候选词的数量和质量也将影响后续的候选窗口数量和选择。因此，在基于上下文预测候选词列表时，所使用的候选词阈值 α 也是影响水印文本质量的重要因素之一。

在每个窗口嵌入 1 bit 水印信息的情况下，即固定 $d=1$ bit，本文采用不同候选词阈值 α 和不同窗口大小 k 生成水印文本，并对水印文本的质量进行评估，实验结果如表 2 所示。从表 2 可以发现，当候选词阈值从 0.010 增加至 0.020 时，水印文本的 PPL 随之下降，窗口大小为 4、5 和 6 的 SS 分别提升了 0.32%、0.46%和 0.60%。这说明随着候选词阈值的增大，低质量的可替换候选词的数量减少，从而使句子质量和相似度整体都得到提升。

表 2　不同候选词阈值和窗口大小下的水印文本质量

α	PPL			SS		
	$k=4$	$k=5$	$k=6$	$k=4$	$k=5$	$k=6$
0.010	70	68	65	91.77%	92.53%	93.28%
0.015	67	66	63	92.05%	92.96%	93.33%
0.020	**65**	**64**	**62**	**92.09%**	**92.99%**	**93.88%**

3.3　嵌入成功率分析

尽管候选词阈值增大，水印文本的质量得到了提升，但候选词数量减少，候选窗口数量将随之减少，会导致窗口压缩编码时，存在一定的概率无法获得水印编码值与待嵌入水印信息值一致的候选窗口，从而出现嵌入失败的情况。

在 α=0.02 的情况下，本文对嵌入过程中成功嵌入水印信息的窗口进行了统计，嵌入成功率如表 3 所示。本文定义的嵌入成功率为嵌入随机水印信息时嵌入成功的窗口数占窗口总数的比例。随着 k 的增大，以及 d 的减小，嵌入成功率随之提升。在 k=6，d=1 bit 时，嵌入成功率达到了 99.97%，仅有极少数的窗口嵌入水印信息失败。

表 3　嵌入成功率

k /个	窗口总数	嵌入成功率	
		$d=1$ bit	$d=2$ bit
4	22 748	99.40%	94.74%
5	18 198	99.82%	97.34%
6	15 165	99.97%	98.57%

在上述实验中，本文方法采用的是固定候选词阈值的方式来获取每个掩码位置的候选词，通过候选词替换原词得到候选窗口和候选句。固定的阈值能保证每个位置上的候选词对其所在的上下文均具有较高的合适度，但不同的位置预测到的候选词数量不同，候选词数量过少时容易导致水印嵌入失败。通过 3.2 节的分析可以发现，降低候选词阈值能够增加候选词的数量，从而增加候选窗口的数量。由于窗口压缩编码方法对窗口进行编码时，编码结果随着窗口内容的改变可能会发生变化，具有一定的随机性。理论上，如果候选窗口数量增加，则会增加窗口的水印编码值的多样性，从而能更好地嵌入指定的水印信息，降低嵌入失败率，甚至达到 100%的嵌入成功率。但随着候选词阈值的降低，意味着更多低质量的词可能被选为候选词，若低质量的候选词用来替换原词生成水印文本，则会导致水印文本的质量的下降。总之，候选词阈值的大小和设置方式会影响水印文本的质量和嵌入成功率。因此，为了对比，本文以 100%的嵌入成功率为目标，将候选词阈值从设置为固定值调整为不确定值进行了相关的实验。本文将该实验方案命名为自适应候选词阈值方式，不同位置获取候选词的阈值随着实际情况自适应变化。

自适应候选词阈值方式不对候选窗口进行过滤，仅获得一个满足水印信息嵌入条件的窗口则完成当前窗口的水印嵌入。具体来讲，当前窗口的水印编码值与待嵌入水印信息不一致时，通过词替换来调整当前窗口的水印编码值，使之能满足水印信息嵌入的条件。对当前窗口中的每个词，根据预测概率对其候选词从大到小进行排序，然后按照预测概率从大到小依次选取一个候选词替换原词得到新的候选窗口。计算候选窗口的水印编码值，若与待嵌入水印信息值一致，则选定该候选窗口来嵌入水印信息；否则，继续选择下一个候选词来替换原词生成下一个候选窗口。在这个过程中，如果生成的候选窗口不满足嵌入水印信息的编码要求，截取候选词的阈值将继续降低，选择下一个预测概率值较低的候选词来替换原词得到新的候选窗口，直到生成水印编码值满足要求的候选窗口，实现水印信息的成功嵌入。

本文设置不同窗口大小 k 和窗口编码长度 d，生成了自适应候选词阈值方式下的相应水印文本。本文从文本质量和嵌入成功率等方面对固定和自适应候选词阈值 2 种方式下的水印文本的性能进行了评估，实验结果如表 4 所示。从表 4 中可以发现，自适应候选词阈值方式实现了 100%的嵌入成功率，且在相同的参数条件下，略微降低了 PPL 的值，获得了比固定候选词阈值下稍好的水印文本质量，但降低了水印文本与原始文本的 SS。这是因为固定候选词阈值方式以语义相似度优先来选择候选窗口，而自适应候选词阈值方式则以词的预测概率优先来选择满足水印编码要求的候选窗口，这种方式下选择的候选词更适合当前上下文，但不一定是语义更接近原词。而 PPL 的计算模型和词预测模型的原理具有一定的相似性，因此，预测概率值更高的词替换原词容易让水印文本的 PPL 更低，但水印文本与原始文本的语义相似度可能会降低。

3.4　水印容量和水印文本质量对比分析

水印容量是评估数字水印方法性能的一个重要

表 4 不同候选词阈值设置方式下的实验结果

方式	k	d = 1 bit			d = 2 bit		
		PPL	SS	嵌入成功率	PPL	SS	嵌入成功率
固定候选词阈值 α=0.02	4	65	92.09%	99.40%	80	87.98%	94.74%
	5	64	92.99%	99.82%	79	88.91%	97.34%
	6	62	**93.88**%	99.97%	72	**90.43**%	98.57%
自适应候选词阈值	4	64	89.98%	**100%**	75	85.01%	**100%**
	5	61	90.82%	**100%**	74	86.06%	**100%**
	6	**58**	91.80%	**100%**	**70**	87.31%	**100%**

指标，指的是水印文本中所能嵌入的最大水印信息量。但由于水印文本的长度不固定，通常一种水印方法的水印容量随着文本长度的变化而变化，因此，在对比实验中，本文使用了嵌入率来衡量水印方法能够嵌入的水印信息量的能力。嵌入率指一个词中平均能够嵌入的水印信息比特数（BPW, bit per word）。

本文选择与经典的基于同义词替换的水印方法[14]和 2 种基于 BERT 的词替换自然水印方法[16-17]进行了对比。每种方法均生成了 10 000 条水印句子。本文统计了所有水印句子的总水印容量、BPW 和 PPL，实验结果如表 5 所示。从表 5 可以看出，文献[14]方法的嵌入率仅为 0.030 6，文献[16-17]方法扩充了可替换词数量，提升了嵌入率，分别达到了 0.150 1 和 0.043 1。文献[17]方法为了保证句子语义变化尽可能小且在提取时生成一致的候选词列表，对候选词进行了严格限制，导致符合要求的可替换词过少，嵌入率提升不明显。与这些方法相比，本文方法的嵌入率最高，能提供更大的水印容量。

表 5 总水印容量和水印文本质量对比

方法	总水印容量	BPW	PPL
文献[14]方法	5 370	0.030 6	**45**
文献[16]方法	26 298	0.150 1	70
文献[17]方法	7 554	0.043 1	47
本文方法（k=6，d = 1 bit）	22 748	0.129 9	62
本文方法（k=6，d = 2 bit）	**45 496**	**0.259 8**	72

在本文方法中，当窗口大小 k =6，d=1 bit 时，嵌入率为 0.129 9；d=2 bit 时，嵌入率提高一倍。随着水印容量的增长，PPL 升高，水印文本的质量有所下降，但 PPL 的增速没有嵌入率高，牺牲了较少的句子质量，嵌入容量得到了更大的提升。相较于文献[14]方法，本文方法基于上下文信息预测获得了更多的可替换候选词，并且设计了灵活的编码方式，可以根据需要选择每窗口嵌入的比特数，大幅度提高了嵌入率。文献[16]方法通过对固定位置上的词生成可替换的候选词并进行替换来实现水印信息的嵌入。固定的位置前后的上下文将在水印嵌入前后保持不变，从而保证水印嵌入和水印提取过程中能够获得一致的候选词列表。固定的位置越多则能够嵌入的水印信息越多，但上下文信息将越少，上下文信息的减少容易导致预测候选词时难以准确预测出适合上下文的候选词。而本文方法设计的水印编码方式不需要考虑水印嵌入和水印提取时词替换位置所在上下文的一致性，能够根据更长的上下文信息来预测候选词。同时，在选择用来替换原词的候选词时，考虑了替换前后句子的语义相似度，优化了候选词的选择。因此，在嵌入相同的水印信息的情况下，本文方法能够比文献[16]方法生成更高质量的水印文本。从表 5 中可以发现，本文方法在 k=6，d = 2 bit 时的 PPL 与文献[16]方法的 PPL 接近，但本文方法的嵌入率要远高于文献[16]方法的嵌入率。本文方法生成的水印文本的 PPL 随着嵌入率的降低而降低。因此，当在相同的文本中嵌入相同的水印信息时，本文方法所生成水印文本的 PPL 将低于文献[16]方法所生成水印文本的 PPL，说明本文方法能提供更高质量的水印文本。

3.5 提取效率对比分析

不同于传统的水印编码方法，本文方法采用了基于文本字符的窗口压缩编码方法，其水印编码结果不依赖于其他信息，水印提取非常简单，极大地降低了时间和空间复杂度，提高了水印提取效率。

本文在型号为 NVIDIA GeForce RTX 3070 的显卡下，对不同方法的水印提取效率进行了对比实验。实验中，记录了每种方法对所生成的 10 000 条水印句子进行水印提取时的总耗时，再根据所嵌入的水印信息比特数计算提取效率，实验结果如表 6 所示。

表 6　　提取效率实验结果

方法	提取效率/(bit·s^{-1})	总耗时/s
文献[14]方法	25.57	210
文献[16]方法	232.7	113
文献[17]方法	0.27	27 903
本文方法（k=6，d =1 bit）	**947.83**	**24**
本文方法（k=6，d = 2 bit）	**1 895.66**	**24**

从表 6 可以看出，与已有同类方法相比，本文方法极大地提升了水印提取效率。当k =6，d =1 bit 时，本文方法的总耗时仅为文献[17]方法的 0.086%，为文献[14]方法的 11.43%。与文献[14, 16-17]方法相比，本文方法的提取效率分别提升了约 37 倍、4 倍和 3 510 倍。提取效率的大幅度提升是因为在提取水印信息时，本文方法只需对水印文本中词的字符编码信息进行简单运算，即可提取出正确的水印信息，算法的时间复杂度是线性的，且空间复杂度极低。而文献[14]基于同义词替换的方法，在提取水印时需要对水印文本进行扫描，通过与同义词库进行比对，识别文本中的同义词，并寻找对应的同义词集进行解码来提取水印信息，水印提取算法比本文方法复杂，耗时多。文献[16]方法在水印信息提取时，需要使用与嵌入时相同的 BERT 模型和一致的参数，利用词预测模型为水印文本中水印信息嵌入位置的词生成候选词列表并进行编码，以此解码水印信息嵌入位置的词编码信息，因此在提取时需要花费较多的时间。文献[17]方法的提取过程需要遍历句子中每个词，加载词预测模型为其生成候选词列表，还需要额外加载语义相似模型对候选词进行排序，提取算法的复杂度最高，需要消耗的时间最多，提取效率最低。经过对比分析可发现，本文方法能够极大地简化水印提取过程，提高水印提取效率，从而大大提高了水印方法的实际应用价值。

4　结束语

本文提出了一种高提取效率的基于词替换的自然语言水印方法。该方法利用基于深度神经网络的预训练语言模型高性能地预测可替换原词的候选词列表，有效地扩充了传统的基于同义词替换的水印方法中的可替换词数量，提高了水印嵌入容量和水印文本质量。在词替换过程中，通过候选句与原句的语义相似度来优化候选词的选择，以降低替换后句子语义的失真。在此基础上提出了一种新的压缩编码方法，在提取时，只需要水印文本和窗口编码长度参数，不需要进行复杂的可替换候选词生成过程以及共享复杂参数，解决了对词预测模型的依赖问题，大大提高了提取效率。实验结果表明，本文方法能够生成高质量水印文本，与已有方法相比，具有更高的水印容量和提取效率，以及较强的实用性。

尽管本文方法具有提取效率高、生成的水印文本质量较好、水印容量大等优势，但是当水印文本遭受攻击导致文本内容发生改变时，窗口的水印编码值可能会发生改变，从而使提取的水印信息有误，即本文方法存在鲁棒性较差的缺点。其次，尽管本文方法所生成的水印文本与原始文本高度相似，但基于上下文预测获得的候选词替换原词，将不可避免地在某些情况下导致水印文本的局部语义偏离原始文本的真实语义。对于完全不能容忍任何语义失真的场景，即使是细微的修改也可能造成原始内容语义上的永久性失真，这时需要设计可逆的自然语言数字水印方法，使从水印文本中提取水印的同时能无损地还原原始文本内容。未来工作将重点研究如何提高词替换水印方法的鲁棒性和原始文本内容的可逆恢复。

参考文献：

[1] THONNARD O, BILGE L, KASHYAP A, et al. Are you at risk? Profiling organizations and individuals subject to targeted attacks[C]//Proceedings of International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2015: 13-31.

[2] WAN W B, WANG J, ZHANG Y M, et al. A comprehensive survey on robust image watermarking[J]. Neurocomputing, 2022, 488: 226-247.

[3] LUO X Y, LI Y X, CHANG H W, et al. DVMark: a deep multiscale framework for video watermarking[J]. IEEE Transactions on Image Processing, 2023, PP(99): 1.

[4] YAMNI M, KARMOUNI H, SAYYOURI M, et al. Efficient watermarking algorithm for digital audio/speech signal[J]. Digital Signal Processing, 2022, 120: 103251.

[5] 何路, 桂小林, 田丰, 等. 自然语言水印鲁棒性分析与评估[J]. 计算机学报, 2012, 35(9): 1971-1982.
HE L, GUI X L, TIAN F, et al. Analyzing and evaluating the robustness of natural language watermarking[J]. Chinese Journal of Computers, 2012, 35(9): 1971-1982.

[6] XIAO C, ZHANG C, ZHENG C X. FontCode: embedding information in text documents using glyph perturbation[J]. ACM Transactions on Graphics, 2018, 37(2):1-16.

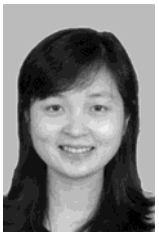
[7] QI W F, GUO W, ZHANG T, et al. Robust authentication for paper-based text documents based on text watermarking technology[J]. Mathematical Biosciences and Engineering, 2019, 16(4): 2233-2249.

[8] YANG X, ZHANG W M, FANG H, et al. Language universal font watermarking with multiple cross-media robustness[J]. Signal Processing, 2023, 203: 108791.

[9] NOZAKI J, MURAWAKI Y. Addressing segmentation ambiguity in neural linguistic steganography[J]. arXiv Preprint, arXiv: 2211.06662, 2022.

[10] VAROL A M. LZW-CIE: a high-capacity linguistic steganography based on LZW char index encoding[J]. Neural Computing and Applications, 2022, 34(21): 19117-19145.
[11] MERAL H M, SANKUR B, ÖZSOY A S, et al. Natural language watermarking via morphosyntactic alterations[J]. Computer Speech & Language, 2009, 23(1): 107-125.
[12] WANG H, SUN X M, LIU Y L, et al. Natural language watermarking using Chinese syntactic transformations[J]. Information Technology Journal, 2008, 7(6): 904-910.
[13] YANG T Y, WU H Z, YI B, et al. Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding[J]. arXiv Preprint, arXiv: 2203.03795, 2022.
[14] WINSTEIN K. Lexical steganography through adaptive modulation of the word choice hash[R]. 1999.
[15] BOLSHAKOV I A. A method of linguistic steganography based on collocationally-verified synonymy[C]//Proceedings of International Workshop on Information Hiding. Berlin: Springer, 2004: 180-191.
[16] UEOKA H, MURAWAKI Y, KUROHASHI S. Frustratingly easy edit-based linguistic steganography with a masked language model[C]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2021: 5486-5492.
[17] YANG X, ZHANG J, CHEN K, et al. Tracing text provenance via context-aware lexical substitution[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2022: 11613-11621.
[18] 武睿峰, 何路, 房鼎益. 自然语言水印隐蔽性自动评测方法[J]. 计算机应用, 2013, 33(12): 3522-3526, 3530.
WU R F, HE L, FANG D Y. Automatic evaluation scheme for imperceptibility of natural language watermarking[J]. Journal of Computer Applications, 2013, 33(12): 3522-3526, 3530.
[19] YANG J L, WANG J M, WANG C K, et al. A novel scheme for watermarking natural language text[C]//Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Piscataway: IEEE Press, 2007: 481-484.
[20] 林建滨, 何路, 李天智, 等. 一种抗攻击的中文同义词替换文本水印算法[J]. 西北大学学报(自然科学版), 2010, 40(3): 433-436.
LIN J B, HE L, LI T Z, et al. An anti-attack watermarking based on synonym substitution algorithm for Chinese text[J]. Journal of Northwest University (Natural Science Edition), 2010, 40(3): 433-436.
[21] ZHENG X Y, WU H Z. Autoregressive linguistic steganography based on BERT and consistency coding[J]. Security and Communication Networks, 2022, 2022: 1-11.
[22] ZHENG X Y, FANG Y R, WU H Z. General framework for reversible data hiding in texts based on masked language modeling[J]. arXiv Preprint, arXiv: 2206.10112, 2022.
[23] CHANG C C. Reversible linguistic steganography with Bayesian masked language modeling[J]. IEEE Transactions on Computational Social Systems, 2023, 10(2): 714-723.
[24] 杨潇, 李峰, 向凌云. 基于矩阵编码的同义词替换隐写算法[J]. 小型微型计算机系统, 2015, 36(6): 1296-1300.
YANG X, LI F, XIANG L Y. Synonym substitution-based steganographic algorithm with matrix coding[J]. Journal of Chinese Computer Systems, 2015, 36(6): 1296-1300.
[25] XIANG L Y, WU W S, LI X, et al. A linguistic steganography based on word indexing compression and candidate selection[J]. Multimedia Tools and Applications, 2018, 77(21): 28969-28989.
[26] YANG Z L, GUO X Q, CHEN Z M, et al. RNN-stega: linguistic steganography based on recurrent neural networks[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(5): 1280-1295.
[27] YU L, LU Y L, YAN X H, et al. MTS-Stega: linguistic steganography based on multi-time-step[J]. Entropy, 2022, 24(5): 585.
[28] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv Preprint, arXiv: 1706.03762, 2017.
[29] HILL J, SIMHA R. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams[C]//Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications. Stroudsburg: Association for Computational Linguistics, 2016: 23-30.
[30] FEDUS W, GOODFELLOW I, DAI A M. Maskgan: better text generation via filling in the__[J]. arXiv Preprint, arXiv: 1801.07736, 2018.
[31] ZHU W, HU Z, XING E. Text infilling[J]. arXiv Preprint, arXiv: 1901.00158, 2019.
[32] LIU Y H, OTT M, GOYAL N, et al. RoBERTa: a robustly optimized BERT pretraining approach[J]. arXiv Preprint, arXiv: 1907.11692, 2019.

[作者简介]



向凌云（1983− ），女，湖南双峰人，博士，长沙理工大学教授、硕士生导师，主要研究方向为信息安全、信息隐藏、数字水印、隐写分析和自然语言处理等。

黄明豪（1999− ），男，湖南邵阳人，长沙理工大学硕士生，主要研究方向为自然语言数字水印和自然语言处理等。

张晨凌（2000− ），男，湖南邵阳人，长沙理工大学硕士生，主要研究方向为自然语言处理等。

杨春芳（1983− ），男，福建莆田人，博士，信息工程大学副教授、博士生导师，主要研究方向为信息隐藏、多媒体智能理解、网络安全等。