专题：网络智能化与生成式人工智能

基于生成式因果语言模型的水印嵌入与检测

刘明录，郑彦，韩雪，袁向阳，邓超
（中国移动通信有限公司研究院，北京 100053）

摘　要：基于人工智能内容生成（AIGC）技术生成文本具有道德、法律的合规性风险，需要对生成文本内容的流通进行规范和监管，因此对 AIGC 生成文本版权保护的迫切需求随之出现。水印技术是目前使用最广泛的数字版权保护方式。提出了一种应用于生成式因果语言模型的生成文本的水印添加技术，采用事中水印嵌入的方式在文本生成过程中隐式地嵌入文本水印特征编码，相较于传统事后水印添加技术对生成文本质量影响小，具有低感知、透明、鲁棒等优点。实验结果表明，提出的水印嵌入策略具有较好的鲁棒性，经过用户一定程度的编辑后仍旧能有效检出文本嵌入水印。与原有生成策略进行对比，所提方法与现有模型耦合度低，无须调整原有模型结构、训练策略、部署方式，不增加原有生成过程计算成本。

关键词：人工智能内容生成；因果语言模型；数字水印；数字版权
中图分类号：TP181
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2023179

Watermark embedding and detection based on generative causal language model

LIU Minglu, ZHENG Yan, HAN Xue, YUAN Xiangyang, DENG Chao
China Mobile Research Institute, Beijing 100053, China

Abstract: Artificial intelligence generated content (AIGC) generated text itself carried moral and legal compliance risks, and the circulation of generated text content need to be regulated. Therefore, there was an urgent need for copyright protection of AIGC generated text. Watermarking technology was currently the most widely used method for digital copyright protection. A watermark embedding technology was proposed for generating text using generative causal language models. An in-process watermark embedding method was adopted, which implicitly embeded text watermark during the text generation process. Compared to traditional post-process watermark embedding technology, it had less impact on the quality of generated text and had advantages such as low perception, transparency, and robustness. The proposed method has low coupling with existing models and can eliminate the need to adjust the original model structure, training strategies, deployment methods, and increase the computational cost of the original generation process. Through experimental results, the proposed watermark embedding strategy has good robustness and can effectively detect text embedded watermarks even after a certain degree of editing by users.

Key words: AIGC, generated causal language model, digital watermark, digital copyright

收稿日期：2023−07−21；修回日期：2023−09−04

0　引言

随着人工智能内容生成（artificial intelligence generated content，AIGC）技术的逐渐成熟，越来越多的 AIGC 模型开始落地商用，其中文本生成式 AI 模型可以基于用户给定的需求描述使用大规模语言模型内嵌的知识和文法表示能力，快速生成自然流畅、符合指令要求的高质量文章，帮助用户高效完成大量文字处理任务。作为一个强大的生产力工具，AIGC 在带来了巨大的商业价值的同时也引入了潜在的使用风险和安全合规问题。因此，如何对 AIGC 生成内容进行版权保护、来源认证、内容追溯是一个亟待解决的问题。

数字产品的版权保护通常使用电子水印技术实现，基于文本内容使用非加密的策略，将与版权相关的水印特征编码嵌入数字产品中，能够在不影响数字产品本身使用的情况下实现对数字产品的版权保护。目前的文本水印嵌入技术方案主要应用对象是人工书写的文本，由于文字书写人员往往并非信息安全专家，难以在书写文字的同时考虑水印添加策略。因此，现有的技术方案是一种事后水印添加技术，需要在给定文本上进行二次处理实现水印的添加，这种方式大大限制了水印的添加手段。而通过 AIGC 完成的文章，文章本身生成自动化、效率高、过程可控，因此可以在文章生成的过程中采用事中水印添加的方式，实现低感知、透明、鲁棒的水印添加。

文本生成的因果语言模型是一种基于深度神经网络的自然语言处理模型，通过对历史文本的建模预测下一个文本标记的概率分布，并根据特定算法对文本标记进行连续决策，直到完成整个文本生成。本文提出了一种可以应用于生成式因果语言模型的文本水印添加技术，不影响用户对文本本身的阅读体验，通过修改文本生成策略上的隐式嵌入文本电子水印。在本文提出的水印嵌入方案中，用户输入文本生成的需求指令以及指令描述（instruction and input），首先生成一个先导文本上下文（prior text context），然后以句子为单位迭代完成整个文本的生成，在生成文本过程中基于本文提出的句子采样策略控制算法生成嵌入水印的子句，逐句生成直到完成整个文本生成过程，将水印序列编码隐式嵌入生成句子序列中，实现用户低感知的文本水印嵌入。通过本案提出的水印提取算法，用户在进行有限的编辑、删除、插入等操作过程中，仍旧能够有效地提取文本水印，从而实现对人工智能生成文本的版权保护。

1　相关工作

1.1　文本水印的研究现状

传统的文本水印添加技术可以主要分为以下几种策略。

文献[1-3]将文本视为一个通过文本行进行排版的图片，从而能够将图像领域成熟的水印技术直接应用于文本领域中。具体地，可以通过嵌入背景图片，改变文字的字体、颜色、字号、间距，修改文字本身的像素特征等手段对原始文本进行数字水印的嵌入。基于图像特征的电子水印极易被用户感知发现，并且在一定程度上对用户的阅读体验造成影响。用户可以通过对原始文本图片的二次截图、格式转换、图像压缩等手段来模糊原有的水印特征，甚至可以通过 OCR 技术对原始文本的字符串进行识别，并直接对原文本内容进行复制，从而彻底消除原文水印。

文献[4-5]基于文本本身的字符串特征和水印位置计算策略，在文本的特定位置对原始文本进行直接修改。具体地，可以利用文本本身的语义特征进行等价代换，如同义词替换、句式替换、标点替换，也可以基于具有掩码字符预测功能的模型（如 Bert），寻找与当前位置词语嵌入表示相近的语义向量词作为替换词，甚至可以通过相近字形引入一些错别字或语法错误实现文本水印添加。采用的文字替换的方案会更改原始文本内

容，甚至导致文本语义的变化，与水印的透明性相违背。语言本身具有多义性和模糊性，一对同义词在不同的语境下不一定能够成为可替换的同义词，受限于自然语言处理技术能力，现有的替换方案往往会导致原始文本阅读流畅度下降，甚至导致文本出现语义偏移和语法错误。

文献[6-7]利用文本字符串、段落序号、页码等文字版式特征，使用零宽度的 Unicode 字符集对上述特征进行编码形成标记字符串，并隐写在文本相对不易感知的隐蔽位置（如标题、页码、目录等），从而在不改变文本内容、不影响用户阅读体验的同时实现数字水印的嵌入。零宽度字符水印添加策略能够保持嵌入水印前后原始文本内容上的一致性。零宽度字符虽然能够在常规阅读软件上隐藏，但是使用专业的软件工具能够将零宽度字符进行可视化展示，并且零宽度字符种类有限且可枚举，用户可以使用简单的程序对隐写字符进行显示和批量删除，从而造成水印的失效。

1.2 生成式因果语言模型

AIGC 技术是指基于人工智能的内容生成技术，使用千亿级的模型参数在海量数据上进行训练，从而使模型能够拟合真实场景的数据分布，在本文中 AIGC 特指利用 AI 技术进行文本生成。因果语言模型（causal language model，CLM）是一种特殊的 AIGC 模型，它通过对前文的输入进行建模，并预测下一个文字的概率分布，通过若干次循环执行上述动作从左至右完成整个文本的生成，典型的因果模型有 ChatGPT[8]、ChatGLM[9]等。一个典型的 CLM 可以使用式（1）进行概括：

$$\mathrm{CLM}(x)=\Sigma\lg P\left(x_i|x_0,x_1,\cdots x_{i-1};\Theta\right),x\in[0,V] \quad (1)$$

$\mathrm{CLM}(x)$ 表示基于该模型的一段文本 x 的出现概率，其中，P 为基于模型输出结果归一化的概率函数，Θ 为模型参数，V 为模型解码空间字典。通常在超大参数 Θ 的情况下，对整个 V 空间进行解码具有较大的计算复杂度，因此一般使用集束搜索技术——beam search[10]进行优化解码优化：beam search 通过维护 n 个输出候选输出结果，每一轮下一词预测时对每一个候选结果分别进行预测，并在全部组合中保留最优的 n 个组合作为本轮预测的输出结果。

在海量训练数据上进行训练的 CLM 通常具有一定程度的语言理解和文本续写能力，但在一些特定任务以及用户的复杂生成需求下往往表现不理想。文本生成式模型如 ChatGPT、ChatGLM 等模型采用了指令微调（instruction supervised finetuning，SFT）[11]算法对原始语言模型进行精调，使语言模型能够理解用户的文本生成指令（prompt），从而生成符合用户预期的文本输出。经过 SFT 后的 CLM 模型的生成过程可以使用式（2）进行概括：

$$\begin{aligned}&\mathrm{CLM}(x)=\\&\Sigma\lg P\left(x_i|x_0,x_1,\cdots x_{i-1};p_0,p_1,\cdots,p_m;\Theta\right),\\&p,x\in[0,V]\end{aligned} \quad (2)$$

其中，$p_0,p_1,\cdots,p_m$ 是用户输入文本生成指令 prompt，在本文中用于进行嵌入水印的生成文本，均为经过 SFT 后的 CLM 进行文本生成。

2 水印嵌入算法

在本文中用户输入文本生成的需求指令以及指令描述（instruction and input）作为文本生成的 prompt 提示，水印嵌入过程分为两个阶段：首先在导引上下文生成阶段生成一个先导文本上下文（prior text context）；然后在水印嵌入生成阶段以句子为单位迭代完成整个文本的生成，在生成文本过程中基于本文提出的句子采样策略控制算法生成嵌入水印的子句，逐句生成直到完成整个文本生成过程，将水印特征编码隐式嵌入生成句子序列中。由于在生成过程中，本文不直接修改最终的生成内容，而是仅对模型输出的候选子句采样的方式嵌入一位水印，因此用户几乎无法感知到生成文本中水印的存在。水印嵌入实施流程如图 1 所示。

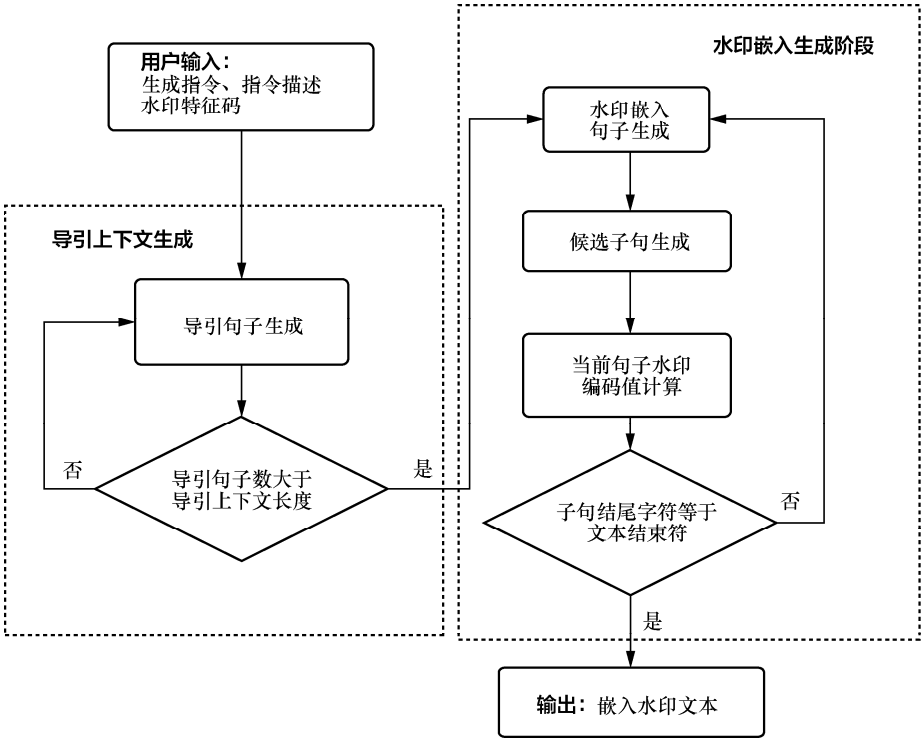


图 1　水印嵌入实施流程

2.1　算法符号定义

水印特征编码 watermark_code 是一个长度为 hash_len 的有限长度列表，在算法的生成和检出过程中水印特征编码需保持不变，一个水印特征编码满足如下特征：

$$\text{watermark\_code}=\left[c_0,c_1,\cdots,c_{\text{hash\_len}}\right] \tag{3}$$

其中，$c_i\in[0,1]$，引导上下文长度 prior_len 引导上下文子句数目。

可读性阈值 λ_{read} 用于调整水印嵌入策略，控制生成文本可读性：

$$\lambda_{\text{read}}\leqslant\lg 1 \tag{4}$$

2.2　引导上下文生成

引导上下文为一段由 prior_len 个子句构成的生成文本，由式（2）可知，在水印嵌入阶段 CLM 基于用户 prompt 进行文本生成，通过深度神经网络对用户 prompt 进行复杂的编码和推理，并生成与 prompt 高度相关的文本输出。但是在水印检测阶段很难直接获得待检测文本的 prompt，因此在生成过程中本文基于用户的 prompt 生成一段无水印的先导文本，从而能够将 prompt 信息直接编码在生成文本中；并在水印检出阶段作为 CLM 的后文生成 prompt，从而能够在没有用户原始 prompt 的情况下对待检测文本进行建模。引导上下文的生成采用传统因果语言模型的生成方式直接生成，引导上下文的生成方式见算法 1。

算法 1　引导上下文的生成方式

输入：文本生成指令 prompt

（1）初始化句子计数 $s_{\text{id}}=0$，引导上下文句子列表 priori_context_sents=[]

（2）基于 beam search 生成子句：sent = top1(beam_search(prompt+prior_context_sents))

（3）$s_{\text{id}}+=1$

（4）将生成句子 s 加入 priori_context_sents 列表末尾

（5）如果 $s_{\text{id}}<$ prior_len 重复执行第（2）、（3）、（4）行

输出：引导上下文 priori_context_sents

2.3 水印嵌入文本生成

水印嵌入生成阶段是在文本中嵌入水印的核心阶段。传统事后水印技术需要直接对最终文本进行修改，这会导致原文语义的漂移，违背了水印透明性的原则。本文基于 AIGC 自动化、高效的文本生成模式，在生成过程中直接生成带嵌入水印的文本。

AIGC 通常基于历史文本对下一个文本标记（token）进行预测，在本文中 AIGC 模型的历史文本为前文所述的引导上下文，在文本生成过程中以子句为生成单位，同时生成多个子句候选，并根据当前位置的水印特征编码值对子句进行采样决策。

假设当前待生成水印嵌入子句句子序号为 $s_{\mathrm{id}}=s_i$，当前水印特征编码 watermark _ code = $\left[c_0,c_1,\cdots,c_{\mathrm{hash\_len}}\right]$，则当前子句待嵌入的一位水印特征编码为：

$$s_{\mathrm{code}}=\mathrm{watermark\_code}\left[s_{\mathrm{id}}\%\mathrm{hash\_len}\right] \quad (5)$$

在水印嵌入候选子句生成中，基于引导上下文作为生成提示，首先采用 beam search 策略进行子句解码搜索，获得 n 个候选子句，然后根据候选子句生成概率、s_{code} 和可读性阈值，从候选子句中选择一个句子作为最终输出，水印嵌入文本生成见算法 2。

算法 2 水印嵌入文本生成

输入：引导上下文句子列表 priori_context_sents = $\left[\mathrm{sent}_0,\mathrm{sent}_1,\cdots,\mathrm{sent}_{\mathrm{prior\_len}-1}\right]$

（1）初始化句子计数 $s_{\mathrm{id}}=\mathrm{prior\_len}$，水印嵌入子句列表 watermark_sents=priori_context_sents

（2）根据式（5）计算当前子句 s_{code}

（3）基于 beam search 生成候选子句：

$\mathrm{sub\_sent}_0,\mathrm{sub\_sent}_1,\cdots,\mathrm{sub\_sent}_n=$

$\mathrm{top}_n\left(\mathrm{beam\_search}\left(\mathrm{watermark\_sents}\right)\right)$

（4）计算候选子句分值：对于子句 s_p，使用式（6）计算概率分值：

$$\mathrm{score}\left(\mathrm{sub\_sent}_p\right)=\frac{\Sigma\lg P\left(x_{pq}|x_{p0},x_{p1},\cdots,x_{pq-1};\mathrm{watermark\_sents}\right)}{\mathrm{len}(\mathrm{sub\_sent}_p)} \quad (6)$$

（5）将分值从高到低排序，不失一般性地假设第一分值、第二分值对应子句为 $\mathrm{sub\_sent}_0$、$\mathrm{sub\_sent}_1$，确定输出嵌入水印子句方式为：

如果 $s_{\mathrm{code}}==1$ and score $(\mathrm{sub\_sent}_1)>\lambda_{\mathrm{read}}$，则 $s=\mathrm{sub\_sent}_1$

否则 $s=\mathrm{sub\_sent}_0$

（6）将生成句子 s 加入 watermark_sents 列表末尾，$s_{\mathrm{id}}+=1$

（7）如果 s 包含结束符则结束，否则重复执行第（2）～（6）行

输出：水印嵌入文本 watermark_sents

在无水印嵌入生成过程中，生成单位为 token，每生成一个 token 会计算候选 token 的分数选择最高的概率，依次循环生成一直到结束符。在水印嵌入文本生成的过程中，对比无水印嵌入过程多了算法 2 的第（5）行：固定生成模型参数使其返回两条候选子句，第（5）行对比两条候选子句的分值，按照概率选择子句，同时第（5）行相对于其他步骤时间复杂度忽略不计，由此可知加入水印嵌入后的算法复杂度不变。

在水印嵌入文本生成过程中，本文仅通过对 CLM 推送的若干候选子句进行采样的方式编码一位水印，相较于基于图像技术的水印嵌入方式，本文的水印嵌入具有极低的感知性，用户无法通过文本本身判断是否有水印以及水印的内容。另外，本文通过子句级别的采样以及可读性阈值 λ_{read} 保证了生成文本的可读性，确保了生成文本的流畅性，并具有较高的水印透明度。最后，本文通过在多个子句的生成过程中对水印特征编码的循环利用，确保了用户在一定程度增删改的基础上仍旧能够实现水印的检出，实现了鲁棒性的水印嵌入。

3　水印检测算法

3.1　算法原理

水印检测算法是水印嵌入过程的逆过程。在水印检测过程中，首先需要进行待检测文本的水印特征编码序列抽取：将待检测文本的头部片段作为引导上下文，将待检测文本的剩余部分视为水印嵌入生成文本，基于 beam search 生成候选子句，并与待检测文本的真实子句进行距离比较。如果真实子句与 Top1 候选子句距离最近，则认为当前待检测文本嵌入的水印特征编码为 0，否则为 1。以子句为单位依次进行遍历，直到获得整个水印特征编码序列。水印检测实施流程如图 2 所示。

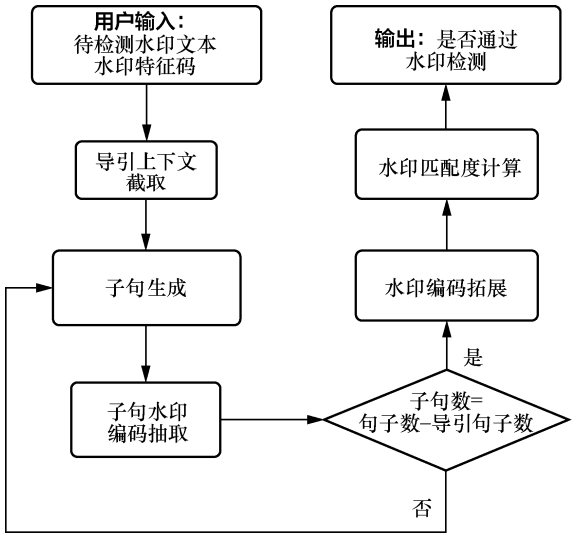


图 2　水印检测实施流程

3.2　水印特征编码抽取

待检测文本的水印特征编码是一个由 0、1 组成的序列，待检测文本的每一个子句可以抽取一位水印特征编码。在水印特征编码抽取阶段，CLM 模型基于历史文本对每一个子句进行生成预测，通过生成的候选子句与原始句子进行文本距离比较，并根据比较的结果确定当前子句的水印特征编码值。文本水印特征编码抽取见算法 3。

算法 3　文本水印特征编码抽取

输入：待检测文本子句列表 article_sents=$[\text{sent}_0,\text{sent}_1,\cdots,\text{sent}_n]$，文本距离比较函数 distance

（1）初始化句子计数 = prior _ len，引导上下文句子列表 priori_context_sents= $[\text{sent}_0,\text{sent}_1,\cdots,\text{sent}_{\text{prior\_len}-1}]$，文本水印特征码列表 textmark_code=[]

（2）获取当前待提取水印特征编码子句 $\text{sent}=\text{article\_sents}[s_{\text{id}}]$

（3）基于 beam search 生成候选子句：

$$\text{sub\_sent}_0,\text{sub\_sent}_1,\cdots,\text{sub\_sent}_n=\text{top}_n\left(\text{beam\_search}\left(\text{priori\_context\_sents}\right)\right)$$

（4）采用式（6）计算候选子句分值，获取分值第一、第二高的子句 sub_sent_0、sub_sent_1

（5）比较 sub_sent_0、sub_sent_1 与 sent 的距离：

如果 $\text{distance}\left(\text{sub}_{\text{sent0}},\text{sent}\right)>\text{distance}\left(\text{sub}_{\text{sent1}},\text{sent}\right)$ 则当前子句水印特征编码 $s_{\text{code}}=1$，否则 $s_{\text{code}}=0$

（6）将 s_{code} 加入 textmark_code 列表末尾，$s_{\text{id}}+=1$

（7）重复执行第（2）～（6）行

输出：待检测文本水印特征编码：textmark_code

在算法 3 中 distance(*,*) 是一个文本距离的比较函数，在后文的实验中本文使用基于编辑距离的距离函数度量文本距离。

3.3　水印特征编码匹配度计算

待检测文本的水印特征编码（textmark_code）与系统的水印特征编码（watermak_code）匹配可以视为两个序列之间的距离匹配，本文使用 Levenstein 距离[12]进行编码的匹配度计算。Levenstein 距离评估了两个序列——a 和 b，a 要做多少个操作可以转换为 b，其中 Levenstein 距离计算包含 3 种操作——增加、删除、替换，在本文中分别对应用户对生成文本的插入新子句、删除子句、修改子句。由于系统的水印特征编码 watermak_code 长度通常远小于待检测文本的水

印特征编码 textmark_code，因此采用循环重复的方式对 watermak_code 进行拓展：

$$\begin{aligned}&watermark\_code = watermark\_code \cdot \\ &len(textmark\_code) / len(watermark\_code)\end{aligned} \tag{7}$$

最后使用 Levenstein 距离计算两个特征编码匹配，采用式（8）计算最终匹配度：

$$sim\_ratio = 1 - \frac{Levenstein\_distance(watermark\_code, textmark\_code)}{\min\left(len(watermark\_code), len(textmark\_code)\right)} \tag{8}$$

4 实验与分析

4.1 数据集与实验设置

为验证文本提出的水印嵌入与检测算法，本文选择使用中文开源大规模语料集 WuDaoCorpora2.0[13] 和 CLUEbenchmark[14]作为算法评测语料，具体如下。

WuDaoCorpora2.0 是由北京智源人工智能研究院发布的大规模自然语言语料集。WuDaoCorpora2.0 包含 200 GB 开源语料，语料中每一个样本带有一种类型标签，本文过滤了标签类型为“问答”的样本，并使用样本的标题作为用户文本生成指令 prompt。

CLUEbenchmark 是一个由开源组织维护的大规模文本语料，语料由多个数据集构成，包含多种自然语言处理子任务，本文使用了其中的问答子任务数据集 baike2018qa。baike2018qa 的每一个样本由一个问题和答案对组成，本文使用问题作为用户文本生成指令 prompt。

在不影响实验结果正确性的前提下，本文使用了启发式的数据清洗规则，删除了数据集中的空行，删除了数据中连续标点符号的空子句。由于本文需要通过截取引导上下文对后文水印嵌入进行检测，因此过滤了子句数量少于 10 个的数据样本。最后受限于生成式因果语言模型庞大的计算开销，本文采用均匀抽样的方式抽取 400 条样本作为本文的水印嵌入与检出的评测数据，实验数据具体统计特征见表 1。

表 1 实验数据及其统计特征

数据集	样本量/个	平均句子数/个	平均 token 量/个
WuDaoCorpora2	200	25	734
CLUEbenchmark	200	31	826

在生成式因果语言模型上，本文使用基于超大规模语料预训练并经过 SFT 的 ChatGLM-6B[15]作为本文的文本生成模型。实验参数配置见表 2。

表 2 实验参数配置

参数名	参数值	备注
prior_len	4	导引句子长度
beam search	6	候选子句数量
top_*k*	90	
重复生成惩罚	0.9	
停止符	‘。’	
水印长度	4、6、8、12	
Lambda_read	0.5、0.3、0.1、0.01	
随机编辑比例	10%、20%、30%	

实验中，本文使用样本的 prompt 作为文本生成指令，采用本文所提出的水印嵌入生成策略生成嵌入水印文本（后文简称水印嵌入回答）作为测试正样本集，使用 CLM 基于 beam search top_1 的无水印嵌入回答（后文简称无水印回答）作为测试负样本集，并使用本文所提出的水印检出策略判断正负样本集的数据是否由水印嵌入的 CLM 模型生成，实验数据样例如图 3 所示。

为评估文本水印嵌入前后的流畅性，本文引入 GPT-2 语言模型[16]计算正负样本的文本困惑度（perplexity PPL）作为文本流畅性评测指标。为评估水印嵌入前后对生成文本一致性的影响，本文采用 BLEU 值比较水印嵌入前后对文本的影响。

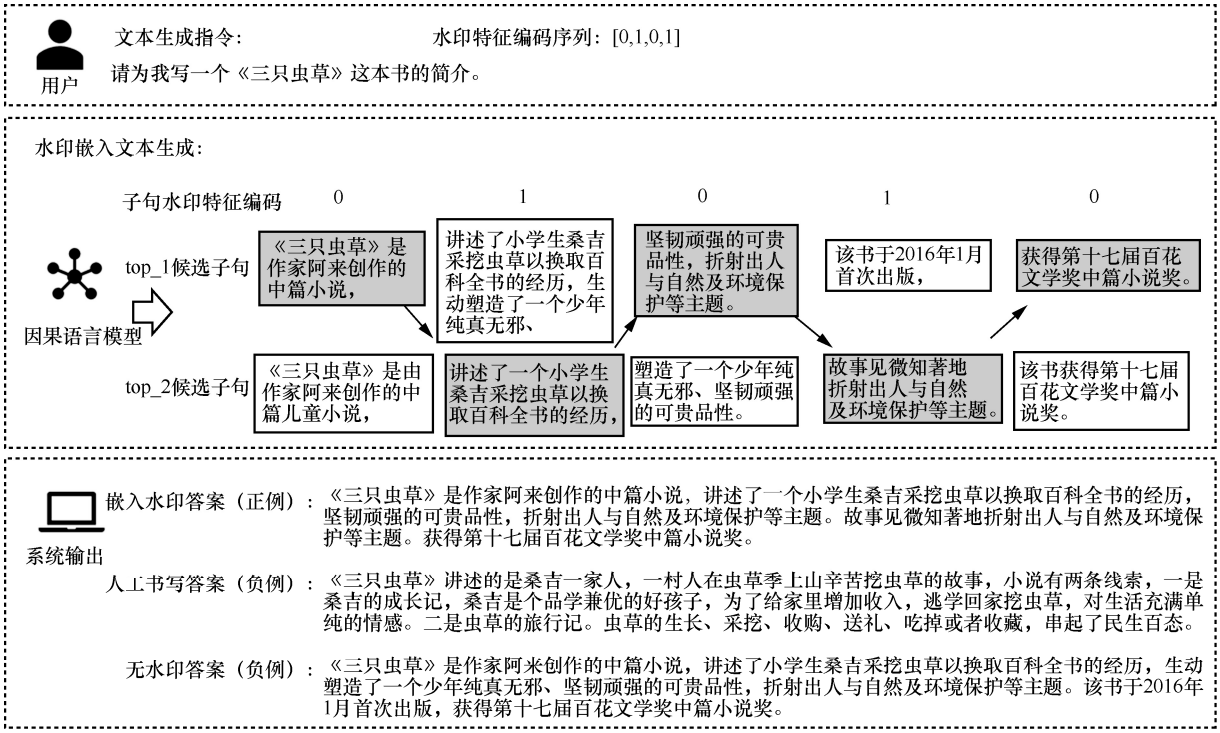


图 3　实验数据样例

4.2　实验结果及分析

水印可读性阈值 λ_{read} =0 时，水印特征编码长度对水印检出影响如图 4 所示。从图 4 可以发现，使用 4 位及以上水印特征编码，已经能够使水印检出的曲线下面积（area under the curve，AUC）值达到 0.98 以上，具有较高的检出性能。同时水印编码长度增长也会提升水印检出正确率。这是由于生成文本嵌入的水印特异性主要由水印特征编码的长度决定，与非水印嵌入文本具有更大的差异性。不同长度水印特征编码长度对生成文本困惑度（PPL）的影响见表 3。

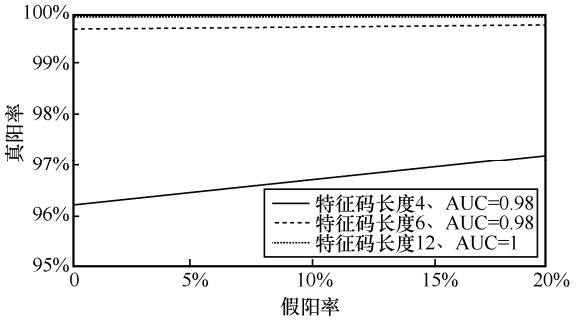


图 4　水印可读性阈值 λ_{read} =0 时，水印特征编码长度对水印检出影响

表 3 比较了不同水印特征编码长度对生成文本困惑度的影响。与无水印回答相比，水印编码

表 3　不同长度水印特征编码长度对生成文本困惑度（PPL）的影响

水印长度	F1	精确率	召回率	PPL 均值	PPL 均值变动百分比
无水印				18.67	
4	98.86%	97.74%	100%	18.79	0.64%
6	99.00%	98.35%	100%	19.92	6.67%
8	99.65%	99.49%	100%	22.83	14.26%
12	99.60%	99.31%	100%	21.78	18.21%

长度较低时 PPL 均值变动较低，更接近原始文本生成策略，采用较高的水印长度时 PPL 均值变动会上升。通常情况下 6 位以下的水印编码已具有良好的效果，从实验结果可知，此时对 PPL 的影响小于 10%。结果表明，本文的水印嵌入策略具有较低的感知度，从人类阅读的角度看几乎不会对可读性造成影响。

水印可读性阈值对水印检出 ROC 曲线如图 5 所示，其比较了水印长度为 6 时，不同 λ_{read} 下水印检出的性能，较低的阈值会使水印检出有更强的鲁棒性，较高的阈值会使生成文本更接近模型最优生成效果，但同时导致更多的子句无法编码正确的水印，整体 AUC 值仍旧能够保持在 0.8 以上，具有较好的检出效果。

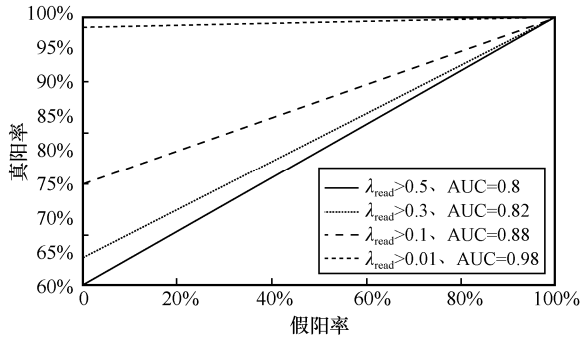


图 5　水印可读性阈值对水印检出 ROC 曲线

水印可读性阈值对生成文本的影响见表 4。为了比较水印嵌入前后对文本困惑度、文本内容一致性的改变，表 7 比较了不同水印可读性阈值 λ_{read} 对生成文本的影响。与无水印回答相比，采用较低的 λ_{read} 时 PPL 均值变动会上升，但上升幅度在 7% 以内。实验结果表明，本文的水印嵌入策略具有较低的感知度，从人类阅读角度看几乎不会对可读性造成影响。为了评估水印嵌入前后对文本内容的影响，本文使用 BLEU 值作为评估手段，将无水印嵌入回答作为参考标准，比较水印嵌入回答的 BLEU 值。从表 4 中可以看出，在文本生成时未使用 λ_{read} 限制时，嵌入水印回答和无水印嵌入的回答的 BLEU 值为 67%，随着使用 λ_{read} 值越高，嵌入水印的回答会越来越与无水印嵌入的回答相似，在 λ_{read} 为 0.5 时，BLEU 的值上升了 15%，上升比例可接受，表明本文的水印嵌入策略具有较高的透明度，采用水印嵌入的文本生成策略对最终的文本生成影响较小。

通常用户在使用机器生成文本时会进行一定程度的编辑，对嵌入水印文本的内容修改会影响水印的检出效果。为模拟用户对于生成文本的修改，本文分别按照 10%、20%、30%的概率对子句进行修改，修改包括插入、删除、替换 3 种操作。水印随机编辑后水印检出结果如图 6 所示，水印随机编辑后水印检出结果见表 5。从图 6 可以看出，一定程度的文本修改（20%以内），本文提出的水印检出算法 F1 仍旧能达到 90.85%；文本修改在 10%以内时，相对于无修改的水印检出策略，本文算法的性能损失小于 1%，该实验结果表明本文所提水印嵌入策略具有比较好的鲁棒性。

表 4　水印可读性阈值对生成文本的影响

λ_{read}	F1	精确率	召回率	PPL 均值	PPL 均值变动百分比	水印回答参考无水印回答 BLEU
无水印				18.67		
0	99.00%	98.35%	100%	19.92	6.67%	67.26%
0.1	85.80%	75.13%	100%	19.45	4.16%	75.11%
0.3	77.91%	63.82%	100%	19.17	2.69%	77.66%
0.5	75.12%	60.15%	100%	18.89	1.18%	80.83%

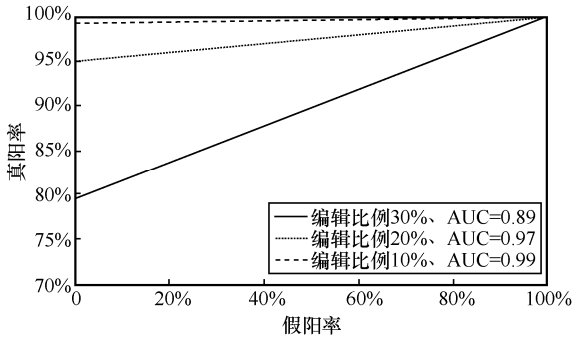


图 6　水印随机编辑后水印检出结果

表 5　水印随机编辑后水印检出结果

随机编辑比例	F1	精确率	召回率
10%	98.34%	96.75%	100%
20%	90.85%	83.25%	100%
30%	69.49%	53.25%	100%

为了检验文本水印嵌入时的复杂度，本文分别在 4 位、6 位、8 位、12 位的水印长度上针对水印嵌入和无水印嵌入的文本生成各 2 000 次，文本水印嵌入复杂度对比见表 6，即显示水印嵌入对比无水印嵌入生成每个 token 的平均时间的比。可以看到，每组有水印嵌入和无水印嵌入生成 token 时间均值差距不大，通过对每组生成时间做 T 检验，各组 P 值远大于 5%无法拒绝原假设，那么生成时嵌入水印和无水印嵌入在时间花销上不存在显著差异。该实验结果表明，本文所提的水印嵌入策略复杂度较低，时间复杂度上与默认文本生成时间复杂度一致。

表 6　文本水印嵌入复杂度对比

对比项	水印嵌入对比无水印嵌入生成每个 token 的平均时间比	T 检验 P 值
4	1.001 4	0.171
6	1.002 4	0.248
8	1.001 48	0.116
12	1.017	0.233

5　结束语

本文提出的一种应用于因果语言模型的生成文本水印嵌入与检测技术，在文本生成过程中自动将水印嵌入生成文本中。实验表明，本文所提水印嵌入技术具有低感知、鲁棒性、透明性等优点。

（1）低感知

通过在文本生成过程中控制采样概率的方式在生成文本中直接嵌入水印，是一种平滑的水印添加策略。在水印嵌入过程中没有使用任何显示或隐式的额外标记添加文本中，用户不易感知到水印的存在。

（2）鲁棒性

语言模型本身是一个基于深度学习网络的平滑概率分布，因此对原始文本进行一定程度的编辑不会导致句子预测概率的大幅度波动。通过多次循环嵌入水印编码来提升水印特征的冗余度，同时采用编辑距离等模糊匹配策略，经过用户一定程度的修改后仍旧能够有效地检测文本水印特征编码。

（3）透明性

在文本生成过程中采用事中水印生成的方案，而非全文完成后对文本进行修改，保证了文本生成前后上下文的一致性。通过可读性阈值保证了文本本身的可读性，水印嵌入不影响文本本身质量，具有较好的透明性。

参考文献：

[1] 刘豪, 孙星明, 刘晋飚. 基于字体颜色的文本数字水印算法[J]. 计算机工程, 2005, 31(15): 129-131.
LIU H, SUN X M, LIU J B. Color-based watermarking algorithm for text documents[J]. Computer Engineering, 2005, 31(15): 129-131.

[2] 王慧琴, 李人厚. 二值文本数字水印技术的研究与仿真[J]. 系统仿真学报, 2004, 16(3): 521-524.
WANG H Q, LI R H. A binary text digital watermarking algorithm[J]. Journal of System Simulation, 2004, 16(3): 521-524.

[3] 周新民, 孙星明, 刘超. 基于汉字结构知识的鲁棒性公开文本水印[J]. 计算机工程与应用, 2006, 42(8): 165-167, 169.
ZHOU X M, SUN X M, LIU C. Robust public text watermarking based on structure knowledge of Chinese characters[J]. Computer Engineering and Applications, 2006, 42(8): 165-167, 169.

[4] 张宇, 刘挺, 陈毅恒, 等. 自然语言文本水印[J]. 中文信息学报, 2005, 19(1): 56-62, 70.
ZHANG Y, LIU T, CHEN Y H, et al. Natural language water-

marking[J]. Journal of Chinese Information Processing, 2005, 19(1): 56-62, 70.

[5] 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.

[6] 傅瑜, 王保保. 文本水印附加空格编码方法的实现及其性能[J]. 长安大学学报(自然科学版), 2002, 22(3): 85-87.
FU Y, WANG B B. Extra space coding for embedding wartermark into text documents and its performance[J]. Journal of Chang'an University (Natural Science Edition), 2002, 22(3): 85-87.

[7] 张震宇, 李千目, 戚湧. 基于不可见字符的文本水印设计[J]. 南京理工大学学报(自然科学版), 2017, 41(4): 405-411.
ZHANG Z Y, LI Q M, QI Y. Text watermarking design based on invisible characters[J]. Journal of Nanjing University of Science and Technology, 2017, 41(4): 405-411.

[8] RADFORD A, NARASIMHAN K. Improving language understanding by generative pre-training[Z]. 2018.

[9] ZENG A, LIU X, DU Z, et al. GLM-130B: an open bilingual pre-trained model[J]. arXiv preprint, 2022, arXiv: 2210.02414.

[10] Wikipedia. Beam search[Z]. 2023.

[11] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[J]. arXiv preprint, 2022, arXiv: 2203.02155.

[12] Wikipedia. Edit_distance[Z]. 2023.

[13] YUAN S, ZHAO H Y, DU Z X, et al. WuDaoCorpora: a super large-scale Chinese corpora for pre-training language models[J]. AI Open, 2021(2): 65-68.

[14] GitHub. CLUE[Z]. 2023.

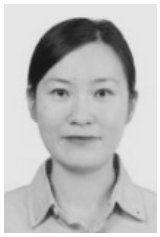
[15] DU Z, QIAN Y, LIU X, et al. GLM: general language model pretraining with autoregressive blank infilling[J]. arXiv preprint, 2021, arXiv: 2103.10360.

[16] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020(33): 1877-1901.

[作者简介]

刘明录（1987– ），男，中国移动研究院人工智能与智慧运营中心算法研究员，主要研究方向为自然语言处理、知识图谱等。

郑彦（1993– ），男，中国移动通信有限公司研究院人工智能与智慧运营中心算法研究员，主要研究方向为大型语言模型及模型的可解释性、公平性。



韩雪（1981– ），女，博士，现任中国移动通信有限公司研究院人工智能与智慧运营中心研究科学家，主要研究方向为 NLP 和多模态融合技术。

袁向阳（1978– ），男，中国移动通信有限公司研究院人工智能与智慧运营中心副总经理，主要研究方向为 BSS、OSS 等 IT 支撑系统及 AI 技术在网络智能化中的应用。

邓超（1978– ），男，中国移动通信有限公司研究院人工智能与智慧运营中心常务副总经理，主要研究方向为人工智能、通信网络智能化、大数据和 IT 技术研发。