

安全性可控的生成式文本隐写算法

梅佳蒙^{1,2}, 任延珍^{1,2}, 王丽娜^{1,2}

(1. 空天信息安全与可信计算教育部重点实验室, 湖北 武汉 430072; 2. 武汉大学国家网络安全学院, 湖北 武汉 430072)

摘要: 生成式文本隐写算法通过对候选池中的单词进行控制性选择映射来隐藏秘密信息, 通常包含3个模块: 文本生成模型、候选池概率分布截断和隐写嵌入算法。由于不同时刻文本生成模型输出的概率分布差异巨大, 现有算法通常采用 $\text{top-}k$ 或 $\text{top-}p$ 对候选池单词的概率分布进行截断, 以减少低概率的生成词, 提高生成文本的安全性。当文本生成模型输出的候选池概率分布过于集中 (over-concentrated) 或过于平坦 (over-flat) 时, 原有的 $\text{top-}k$ 或 $\text{top-}p$ 截断方式不足以应对概率分布的变化, 容易产生概率较低的词或忽略概率较高的词, 导致生成文本的安全性指标出现异常。针对此类问题, 提出了安全性可控的生成式文本隐写算法, 在候选池中根据秘密信息控制性选择生成词时, 所提算法基于困惑度和 KL 散度的参数约束, 动态进行候选池概率分布的截断, 使候选池中所有单词都满足参数约束, 提高了生成文本的安全性。实验结果表明, 所提算法生成的隐写文本困惑度和 KL 散度可控; 在相同 KL 散度情况下, 生成文本的困惑度较现有算法下降最高达 20%~30%; 可以同时控制困惑度和 KL 散度, 在指标合理的情况下, 使生成的文本同时满足困惑度和 KL 散度两个指标。在使用 3 种文本隐写分析算法检测生成的隐写文本时, 检测准确率均在 50% 左右, 表现出很好的统计安全性。

关键词: 生成式文本隐写算法; 算术编码; 安全性可控; 候选池截断

中图分类号: TP37

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2022039

Generation-based linguistic steganography with controllable security

MEI Jiameng^{1,2}, REN Yanzhen^{1,2}, WANG Lina^{1,2}

1. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan 430072, China

2. School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

Abstract: Generation-based linguistic steganography hides secret information through controllable modification and mapping of words in the candidate pool. It usually consists of three parts: text generation model, candidate pool probability distribution truncation and steganographic embedding algorithm. Due to the huge difference in the

收稿日期: 2022-01-12; 修回日期: 2022-04-19

通信作者: 任延珍, renyz@whu.edu.cn

基金项目: 国家自然科学基金 (61872275, 62172306); 湖北省重点研发计划 (2021BAA034, 2020BAB018)

Foundation Items: The National Nature Science Foundation of China (61872275, 62172306), The Key R&D Program of Hubei Province (2021BAA034, 2020BAB018)

引用格式: 梅佳蒙, 任延珍, 王丽娜. 安全性可控的生成式文本隐写算法[J]. 网络与信息安全学报, 2022, 8(3): 53-65.

Citation Format: MEI J M, REN Y Z, WANG L N. Generation-based linguistic steganography with controllable security[J]. Chinese Journal of Network and Information Security, 2022, 8(3): 53-65.

probability distribution of the text generation model outputs at different times, existing algorithms usually use top- k or top- p methods to truncate the probability distribution of words in the candidate pool to reduce the low-probability generated words and improve the security of the generated text. When the probability distribution of the candidate pool output by the text generation model is over-concentrated or over-flat, the original top- k or top- p truncation method will be not enough to cope with the change of the probability distribution, and it is easy to generate low-probability words or ignore high-probability words. This will lead to abnormal security metrics of the generated text. To address these problems, a generation-based linguistic steganography with controllable security was proposed. When selecting generated words with controllability in the candidate pool according to secret information, the proposed algorithm was based on the parameter constraints of perplexity and KL divergence. The truncation of the candidate pool probability distribution made all words satisfy the parameter constraints, which improved the security of the generated text. Experiment results showed that the perplexity and KL divergence of the steganographic text generated by the proposed algorithm are controllable. Under the same KL divergence, the perplexity of the text generated by the proposed algorithm is reduced by up to 20%~30% compared with the existing algorithm. This algorithm could control the perplexity and KL divergence at the same time, and make the generated text satisfy both perplexity and KL divergence when the indicators are reasonable. When using the three text steganalysis algorithms to detect the generated steganographic text, the detection accuracy is about 50%, showing excellent statistical security.

Keywords: generation-based linguistic steganography, arithmetic coding, controllable security, the truncation of candidate pool

0 引言

文本是人类使用最早、使用频率最高的通信方式,在长期的历史发展中,形成了复杂的语义语法规则,保证了信息传递的准确性和有效性。文本内容的语义歧义和信息冗余较少,一个字的差别就会导致截然不同的语义。尽管与其他的多媒体载体相比,以文本为载体的隐蔽信道隐写容量较小,但文本作为最基本的信息交流手段,使用频率高,具有在隐蔽通信中信息无损传输的优势,已成为信息隐藏技术新的优质载体。

传统的文本隐写技术^[1-4]通常是基于文本内容或文本表现形式等信息的编辑修改方法,在嵌入容量和安全性上有明显不足。生成式文本隐写算法利用输出的概率分布来建立生成词与待嵌入秘密信息之间的映射关系,完成秘密信息的嵌入,具有文本语言模型保持一致的特性,得到了广泛研究。基于生成模型的文本隐写方法包含两大类:基于统计模型的文本隐写和基于深度神经网络的文本隐写。

早期的生成式文本隐写算法利用统计模型,如马尔可夫链,对特定语料库进行建模。文献[5-7]利用马尔可夫链的无记忆性和时不变性对随机过

程进行建模,并提出不同的转移状态映射了不同的秘密信息。Yang 等^[8]提出在语言模型输出的概率分布和秘密信息之间使用哈夫曼编码的映射方式,提升了生成文本的安全性。

基于深度神经网络的生成式文本隐写算法常用的语言模型有基于循环神经网络^[9](RNN, recurrent neural network)的生成模型、基于变分自编码器^[10](VAE, variational auto-encoder)的生成模型、基于 Transformer^[11-13]的生成模型、基于生成对抗网络^[14](GAN, generative adversarial network)的生成模型等。此类隐写模型的基本原理如图 1 所示,包含文本生成模型、候选池概率分布截断和映射嵌入算法 3 个关键部分。在生成一段文本的过程中,需要将秘密信息映射嵌入每个生成词中。这些模型生成的隐写文本在隐蔽性和统计安全性的各项指标(如 KL 散度、困惑度)上明显优于传统的语言模型。Fang 等^[15]使用了块状编码,对候选池预先分块(block-based)建立生成词与秘密信息之间的映射关系,语言模型上采用长短期记忆神经网络(LSTM, long short-term memory)。Yang 等^[16]基于双层 LSTM 语言模型建立了一个生成式文本隐写对话系统。Yang 等^[17]使用 LSTM 语言模型,利用哈夫曼编码^[18]在隐写

容量得到保证的情况下大幅降低了生成文本的困惑度 (perplexity), 使用哈夫曼编码时需要将候选池中的单词进行降序排列, 选择前 k 个单词作为新的候选池, 建立与秘密信息的映射关系。Dai 等^[19]在 Yang 等^[17]的基础上, 使用

GPT-2-117M 预训练语言模型^[12-13,20], 提出了 patient-Huffman 的映射嵌入算法, 牺牲隐写容量换取了部分统计安全性。Ziegler 等^[21]和 Shen 等^[22]在映射嵌入算法上选择了算术编码^[23], 大幅提升了生成文本的安全性。

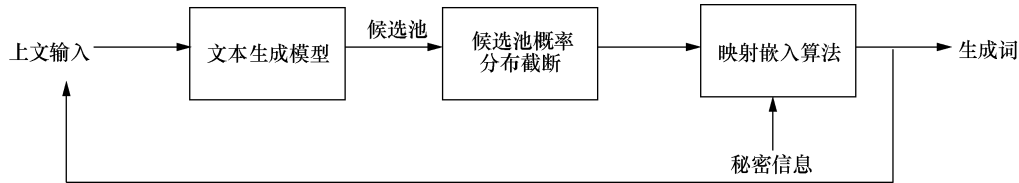


图1 生成式文本隐写模型基本流程
Figure 1 The process of generation-based linguistic steganography model

生成式文本隐写算法中的候选池截断策略对生成文本的隐蔽性、容量和统计安全性有显著的影响。现有方法的候选池截断策略通常采用 $\text{top-}k$ ^[24-25]或 $\text{top-}p$ ^[26]截断。固定数量的 $\text{top-}k$ 或固定概率的 $\text{top-}p$ 在待生成词候选池的概率分布过于集中 (over-concentrated) 或者过于平坦 (over-flat) 时, 会导致生成词的隐蔽性或者统计安全性产生异常。当候选池词概率分布过于集中时, 映射嵌入算法强制选择概率较低的词, 使困惑度增加; 而过于平坦时, 有些概率并不低的词不在新的候选池内, 使前后两个候选池的概率分布差异过大, 导致 KL 散度增加。

本文针对以上问题, 考虑到不同生成词时刻神经网络语言模型输出概率分布之间的差异, 提出安全性可控的生成式文本隐写算法, 该算法基于困惑度和 KL 散度的参数约束, 生成过程直接受控于评价指标 (困惑度和 KL 散度), 自适应地选择合适的 k 值, 使生成文本安全性可控。并且, 本文算法能够兼顾困惑度和 KL 散度两个指标, 使生成文本拥有更好的隐蔽性和统计安全性。

1 相关工作

1.1 文本隐写算法的安全性

文本隐写算法的安全性是指生成的文本不被检测出存在隐写信息的性能, 包括统计安全性和隐蔽性, 统计安全性使用 KL 散度 (KL-divergence) 衡量, 隐蔽性使用困惑度衡量。KL 散度表示两个不同分布之间的差异, 在衡量生成文本质量时表示原有语言模型生成文本 (cover) 的概率分布 $P(\text{cover})$ 和具有嵌入秘密信息语言模型生成的文

本 (stego) 的概率分布 $P(\text{stego})$ 之间的差异。KL 散度定义如式(1)所示。

$$D(P_{\text{cover}} \parallel P_{\text{stego}}) = \sum_{x \in S} P_{\text{cover}}(x) \cdot \log \frac{P_{\text{cover}}(x)}{P_{\text{stego}}(x)} \quad (1)$$

$P(x)$ 是输入单词的联合概率分布, 待输出单词的条件概率分布是 $P(x_{i+1} | x_1, x_2, \dots, x_i)$ 。第 $i+1$ 个单词是取决于前 i 个词的结果。当 $D(P_{\text{cover}} \parallel P_{\text{stego}}) < \varepsilon$ 时, 表示该隐写算法是 ε 不可感知的; 当 $D(P_{\text{cover}} \parallel P_{\text{stego}}) = 0$ 时, 表示该文本隐写算法是绝对不可感知的。

困惑度是 NLP 领域中评价生成文本隐蔽性的指标, 其定义为每个单词在数据集上的平均 $\log$ 概率, 如式(2)所示, 反映了语言模型生成文本的概率分布和训练文本概率分布之间的统计性差异。低困惑度意味着语言模型生成的文本与训练文本差距小, 生成的文本更接近于自然文本。

$$\text{perplexity} = 2^{\frac{1}{n} \log p(s)} \quad (2)$$

根据 Yang 等^[10,17]的研究发现, 语言模型在条件概率编码的框架下, 随着嵌入容量的增加, 生成文本的困惑度会增加; 隐写文本的隐蔽性指标困惑度和统计安全性指标 KL 散度相互矛盾, 两个指标不可能同时达到很好的效果。

1.2 现有的候选池截断策略

主流生成式文本隐写技术需要对各种语言模型的输出结果进行归一化, 得到待生成词候选池的概率分布, 常见的候选池概率分布如图 2 所示, 图中单个单词的生成概率逐渐降低。文本生成模

型输出的候选池概率分布中, 包含数千甚至上万个候选词, 根据长尾效应, 大部分的单词概率偏低, 若映射嵌入时选择了这些词, 将会严重影响生成文本困惑度, 因此需要根据一定的规则对候选池概率分布做截断, 去除分布中概率非常低的词。如何做截断适当去除这些词是生成式文本隐写算法中的重要问题, 现有方法均采用候选池截断方法, 主要包括 $\text{top-}k$ ^[24-25] 和 $\text{top-}p$ ^[26] 两大类。

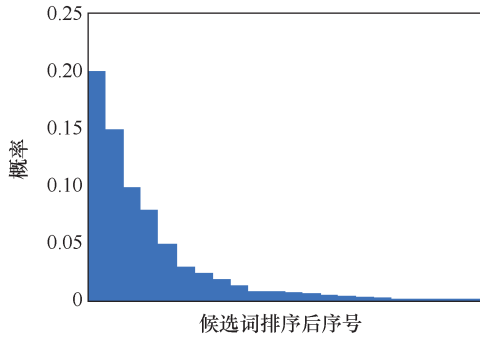


图 2 常见的候选池概率分布

Figure 2 The probability distribution of original candidate-pool

1.2.1 $\text{top-}k$ 截断

$\text{top-}k$ 截断指每次生成单词的时候, 根据神经网络语言模型输出的概率分布, 选取排序后的前 k 个单词作为新的候选池。Ziegler 等^[21]在不同生成时刻使用相同的 k 值进行截断, 生成新的待生成词候选池, 在进行生成式文本隐写时, 采用固定的 k 值, 在 $k=300, 600, 900$ 的情况下完成实验。但这样忽视了候选池概率分布的差异性, 虽然相比无截断, $\text{top-}k$ 截断生成的单词性能在困惑度和 KL 散度上有不少提升, 但由于 k 值固定, 导致生成文本的困惑度和 KL 散度性能较弱。

$\text{top-}k$ 截断的关键在于对 k 值的选择, 但是上下文不同时, 神经网络语言模型输出的候选池概率分布不会相同, 相同的 k 值会忽视概率分布的差异。当 k 很大时, 某些情况下会生成概率过低的词; 当 k 很小时, 某些情况下会忽略概率不低的词, 使生成文本的困惑度和 KL 散度受到影响。

1.2.2 $\text{top-}p$ 截断

$\text{top-}p$ 截断又名核采样 (nucleus sampling), 其关键是选取神经网络语言模型输出的概率分布中前 k 个单词作为新的候选池, k 是单词概率之和大于目标值 p 的最小值。给定输出的概率分布

$P(x|x_{1:i-1})$, 单词集为 V , $\text{top-}p$ 截断的候选池为 $V^{(\text{top-}p)} \subset V$ 的满足条件的最小子集, 如式(3)所示。

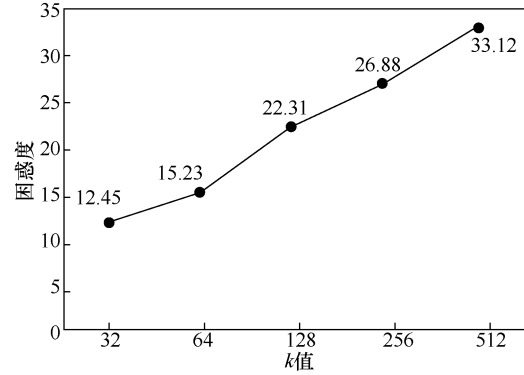
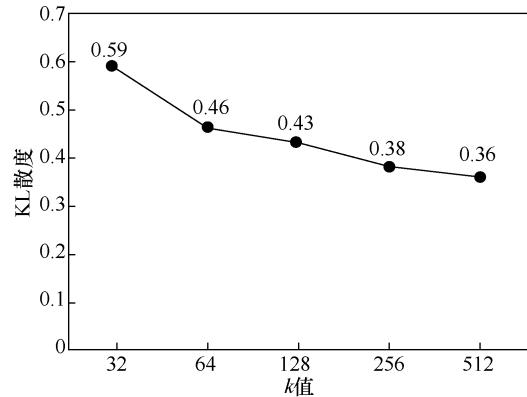
$$\sum_{x \in V^{(\text{top-}p)}} P(x|x_{1:i-1}) \geq p \quad (3)$$

Shen 等^[22]在 Ziegler 等^[21]工作的基础上, 提出了动态自适应的 $\text{top-}k$ 截断方案, 以 $\text{top-}p$ 的算法控制候选池前 k 个词概率分布之和, 参数 k 由参数 δ 决定, 如式(4)所示, 随着候选池概率分布的变化, $\text{top-}k$ 的 k 值动态调整, 有效去除候选池中的低概率单词, 提高嵌入容量和统计安全性。

$$K_t = \min(\{K | \sum_{y' \in K(y < t)} P_{\text{LM}}(y' | y < t) \geq 2^{-\delta}\}) \quad (4)$$

1.3 候选池截断对生成文本的影响

在其他条件相同的情况下, 随着 k 增加, KL 散度的值会减小, 统计安全性更好, 同时困惑度的值会增大, 隐蔽性也就更差。具体结果如图 3、图 4 所示。

图 3 基于算术编码的隐写文本在不同 k 值情况下的困惑度
Figure 3 Comparison of Arithmetic coding and Huffman coding on the perplexity when the k value is the same图 4 基于算术编码的隐写文本在不同 k 值情况下的 KL 散度
Figure 4 Comparison of Arithmetic coding and Huffman coding on the KL-divergence when the k value is different

1.4 现有截断策略存在的问题

$\text{top-}p$ 截断后的候选池只包含原分布中很小一部分的单词，但这部分单词所占的概率和是原分布的 90% 以上，该方法较好地解决了 $\text{top-}k$ 截断的问题，大多数情况下能够忽略概率较低的单词，降低生成文本的困惑度，同时不会过于影响 KL 散度。但在一些极端情况，如待生成词候选池概率分布过于集中或者过于平坦时， $\text{top-}p$ 截断容易生成概率较低的词或忽略概率不低的词，导致生成文本的困惑度或 KL 散度出现异常。式(4)中 k 值是候选池中，概率之和大于目标值 p 的最小 k 值，此目标值与不可感知性，包括隐蔽性——困惑度，统计安全性——KL 散度没有直接的关系，因此采用固定 p 值不能很好地控制生成隐写文本的安全性。

图 5 和表 1 是候选池概率分布较为平坦的一个实例，图 5 是表 1 候选池单词概率的柱状图，表 1 则按照概率大小的顺序，列出了部分单词的概率、困惑度和 KL 散度。上文输入为“Washington received his initial military training and the regiment lost its”。该候选池中单词的最大概率为 0.036，后续单词概率分布平坦。按照现有算法 stego-SAAC^[22]处理， k 在 $\delta=0.1$ 时为 144，考虑到原候选池概率分布尾部的单词概率差距很小，还有一部分单词应被划分到候选池中，否则会降低生成文本的 KL 散度。

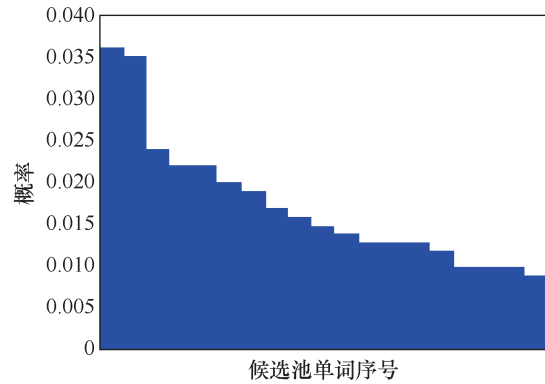


图 5 候选池概率分布过于平坦的实例
Figure 5 The example of probability distribution of the candidate pool is too flat

图 6 和表 2 是在候选池概率分布过于集中时候选池概率分布较为集中的一个实例，图 6 是表 2 的候选池概率的柱状图，表 2 是候选池中部分单词对应的困惑度和 KL 散度，上文输入为“Washington received his initial military training Eventually, Washington included his memoir”。此候选池中，第一个单词的概率为 0.746，占比突出，前 4 个单词的概率占比超过了 94%。针对这种概率分布的情况，按照现有算法 stego-SAAC^[22]处理，在其控制的变量 $\delta=0.05, 0.1, 0.15$ 时，候选池截断的 k 值分别为 4, 7, 20，截断后候选池存在较多困惑度超过 300 的单词，但其 KL 散度并无明显提升，若在嵌入编码时选择了这些单词，则会导致生成文本的困惑度增加。

表 1 候选池概率分布过于平坦时各个单词的 KL 散度和困惑度
Table 1 KL divergence and perplexity of each word when the probability distribution of the candidate pool is too flat

单词序号	单词	概率	困惑度	KL 散度
1	command	0.036	28.310	4.796
2	first	0.035	42.440	3.806
3	way	0.024	44.488	3.398
4	commander	0.022	49.481	3.091
5	headquarters	0.020	52.382	2.840
6	battle	0.019	60.474	2.645
7	reg	0.017	61.558	2.482
8	footing	0.016	67.646	2.355
9	last	0.015	69.348	2.240
10	ability	0.014	75.906	2.142
11	commanding	0.013	76.047	2.053
12	original	0.013	79.395	1.976
...	...	...	...	...

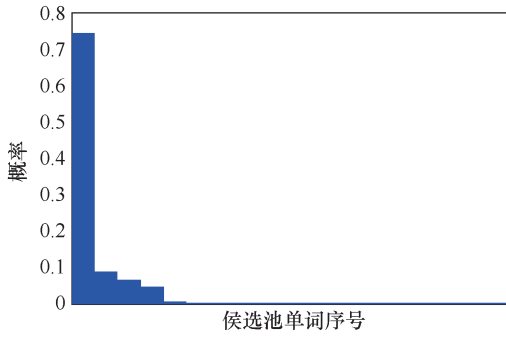


图 6 候选池概率分布过于集中的实例

Figure 6 The example of probability distribution of the candidate pool is too concentrated

表 2 候选池概率分布过于集中时各个单词的 KL 散度和困惑度

Table 2 KL divergence and perplexity of each word when the probability distribution of the candidate pool is too flat

单词序号	单词	概率	困惑度	KL 散度
1	s	0.746	11.248	0.423
2	,	0.089	13.695	0.261
3	in	0.063	20.257	0.140
4	\	0.049	115.357	0.063
5	of	0.009	329.106	0.050
6	The	0.003	330.929	0.046
7	and	0.003	437.421	0.041
...	...	...	...	...

2 安全性可控的生成式文本隐写算法

针对待生成词候选池过于平坦或集中所导致生成词指标异常的问题, 本文提出安全性可控的生成式文本隐写算法, 以文本隐写的安全性评价指标 KL 散度和困惑度作为候选池截断的依据, 提出 $\text{top-controllable } k$ (top-ck) 截断策略, 提升生成文本的安全性。

2.1 算法总体思路

安全性可控的生成式文本隐写算法基于 Transformer 的文本生成模型, 根据上文输入生成下文候选池概率分布, 使用 $\text{top-controllable } k$ (top-ck) 截断方案, 并根据困惑度可控或 KL 散度可控的参数阈值, 动态进行候选池截断, 然后采用基于算术编码的映射嵌入算法, 生成特定的单词以嵌入秘密信息。接下来, 2.2 节详细介绍本算法中的 top-ck 截断方案, 2.3 节介绍映射嵌入算法的嵌入和提取过程。

2.2 top-ck 截断方案

2.2.1 困惑度可控的截断策略 (top-ckppl)

设定困惑度阈值上界 ppl_{thr} , 在候选池的概率分布中, 计算在不同 k 值截断的情况下, 候选池中概率最小单词的困惑度, 使用二分搜索法, 根据阈值 ppl_{thr} 找到满足条件的最大值 k_t , 如式(5)所示。新生成的候选池中, 限定每个单词的困惑度小于阈值 ppl_{thr} 。

$$k_t = \max(\{k \mid \text{perplexity}(v_k \mid y_{t-1}) \leq \text{ppl}_{\text{thr}}, k \in [2, |V|]\}) \quad (5)$$

其中, $\text{perplexity}(v_k \mid y_t)$ 表示在已生成文本序列 y_{t-1} 的条件下, t 时刻根据选择候选池中第 k 个词 v_k 时的困惑度; ppl_{thr} 表示设定的困惑度阈值; $|V|$ 表示候选池中单词的数量。在获取到截断后的候选池概率分布后, 根据待嵌入的秘密信息, 使用基于算术编码的映射嵌入算法生成对应的单词。

2.2.2 KL 散度可控的截断策略 (top-ckkl)

在生成式文本隐写算法中, 统计安全性使用 KL 散度表示, 最理想的情况下, 隐写生成文本的分布与自然文本分布之间的 KL 散度为 0, 没有差异。从信息论的角度出发, 通常会计算生成的隐写文本 P_{stego} 和语言模型正常生成的文本 P_{cover} 之间的 KL 散度, 表示隐写算法的统计安全性。

根据上文结论可知, 生成式文本隐写算法中的 k 值越大, 隐写文本分布与原始语言模型分布间的差异越小, KL 散度越小, 统计安全性越高。由于 KL 散度这种递减的趋势, 首先设置 KL 散度阈值上界 KL_{thr} , 之后在候选池的概率分布中, 计算在不同 k 值情况下候选池概率分布与原始概率分布之间的 KL 散度, 然后使用二分搜索找到满足条件式(6)的最小 k 值 k_t 。

$$k_t = \min(\{k \mid D(P_{\text{stego}}(\text{top } k) \parallel P_{\text{LM}}) \leq \text{KL}_{\text{thr}}, k \in [2, |V|]\}) \quad (6)$$

其中, $P_{\text{stego}}(\text{top } k)$ 表示经过 $\text{top-}k$ 截断后的概率分布, P_{LM} 表示原始的概率分布, 两者之间的 KL 散度需要小于设定的阈值 KL_{thr} , 选择满足此条件的最小的 k 值。并且, 困惑度可控和 KL 散度可

控的截断策略可以同时使用，在阈值合理设置的情况下，能够同时兼顾两个安全性指标。

2.3 嵌入和提取过程

2.3.1 嵌入

本文生成式文本隐写算法嵌入过程如图7所示，其简要示例如图8所示，输入秘密信息比特序列，经GPT-2文本生成模型处理后产生概率分布，然后使用top-ckkl或top-ckppl截断策略，并使用算术编码嵌入秘密信息，输出隐写后文本。首先定义算术编码精度，该值决定了算术编码计算过程中的最小区间，假设算术编码精度为 r （通常 $r=16$ ），即算术编码最小区间为 $\frac{1}{2^{16}}$ ， t 时刻的区间范围为 $[L_t, H_t]$ ，当 $t=1$ 时，概率区间为 $[0, 2^{16}]$ ，当时刻为 t 时，语言模型输出的单词序列，根据设定的困惑度或者KL散度的阈值，在经过动态top- k 截断后，按概率降序排列为 $\{y_1, y_2, \dots, y_k\}$ ，对应的概率排序为 $\{P(y_1), P(y_2), \dots, P(y_k)\}$ 。

$$l_j = (H_j - L_j) \sum_{i=1}^{j-1} P(y_i)$$

$$h_j = (H_j - L_j) \sum_{i=1}^j P(y_i) \quad (7)$$

假设待嵌入的秘密信息比特序列为 m ，嵌入步骤如下。

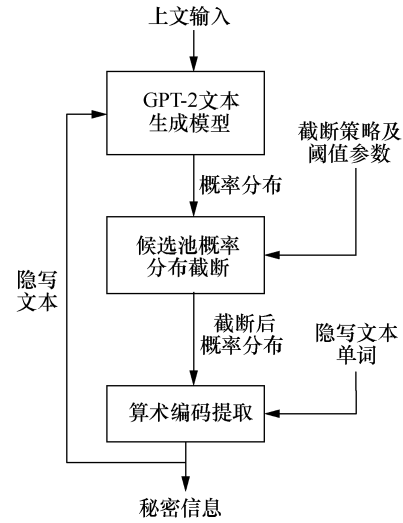


图7 生成式文本隐写算法嵌入过程

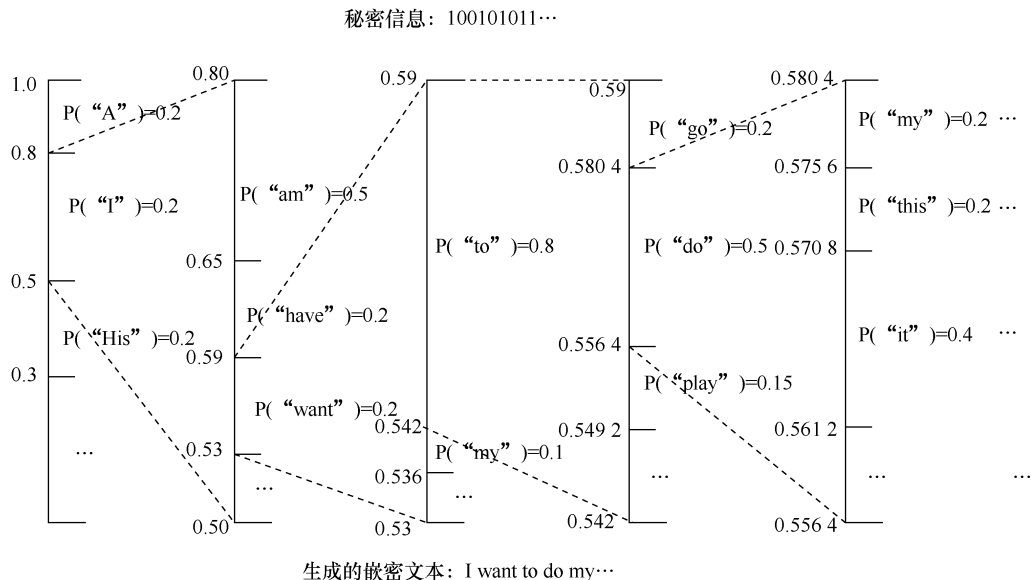
Figure 7 The embedding process of generative text steganography algorithm

1) 语言模型根据上下文的信息得到当前时刻输出的概率分布，此概率分布在做top-ckkl或top-ckppl截断后生成新的概率分布并得到当前的 $[L_t, H_t]$ ，然后根据式(7)得到每一个单词符号 y_j

的概率区间范围 $[\frac{l_j}{2^r}, \frac{h_j}{2^r}]$ 。

2) 根据待嵌入的秘密信息 m 的序列，读取 r 比特信息（通常为16 bit） $m_1 m_2 \dots m_n$ ，转换为十进制小数 M ， $M \in [0, 1)$ 。

3) 寻找对应的单词符号 y_j ，使 M 满足



生成的嵌密文本: I want to do my...

图8 生成式文本隐写算法嵌入过程示例

Figure 8 The example of the embedding process of generative text steganography algorithm

$\frac{l_j}{2^r} \leq M \leq \frac{h_j}{2^r}$, 则当前生成的单词符号为 y_j , 并将区间范围 $[L_t, H_t)$ 更新为 $[l_j, h_j)$, 进行下一步操作。

4) 如果 L_t 和 H_t 的二进制数前 n 位相同, 则表示当前的生成词已经嵌入 n 位信息 $m_1 m_2 \cdots m_n$, 更新 L_t , 秘密信息左移 n 位, 右侧补 0, 更新 H_t , 秘密信息左移 n 位, 右侧补 1。如果 L_t 和 H_t 前 n 位没有对应相同的, 则左移 n 位再次匹配, 如果仍然没有相同的, 则需要跳过进行下次操作, 本次操作并不嵌入秘密信息。

5) 重复步骤 1)~步骤 4), 直到完成秘密信息的嵌入。

2.3.2 提取

提取是嵌入的逆向操作, 输入的是隐写生成文本序列 $y_1 y_2 y_3 \cdots y_n$, 输出秘密消息。本文生成式文本隐写算法提取过程如图 9 所示。

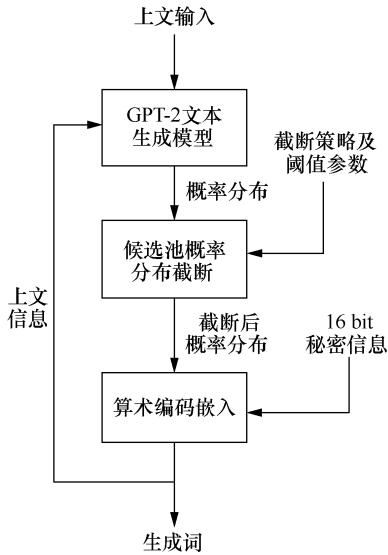


图 9 本文生成式文本隐写算法提取过程
Figure 9 The extraction process of generative text steganography algorithm in this paper

本文生成文本隐写算法提取步骤如下。

1) 在有语言模型和上文信息的前提下, 获取当前时间 t 的语言模型输出的概率分布, 之后根据设定的指标阈值获取动态 top-ckkl 或 top-ckppl 截断后的概率分布。

2) 根据动态 top-ck 截断后的概率分布。获取当前状态下的 $[L_t, H_t)$, 在此基础上根据式(7)计算得出每一个单词 y_j 映射的概率区间 $[\frac{l_j}{2^r}, \frac{h_j}{2^r})$ 。

3) 在时间 t 输入的单词为 y_t , 那么对应的区间范围为 $[L_t, H_t)$, 并将区间范围更新为 $[l_t, h_t)$, 对应的概率区间为 $[\frac{l_t}{2^r}, \frac{h_t}{2^r})$ 。

4) 如果 L_t 和 H_t 的二进制数前 n 位相同, 则可以对应概率区间的二进制比特数据, 提取出前 n 位比特流, 作为提取出的秘密信息。更新 L_t , 秘密信息左移 n 位, 右侧补 0, 更新 H_t , 秘密信息左移 n 位, 右侧补 1。如果 L_t 和 H_t 的二进制数前 n 位没有对应相同的, 则跳过此单词, 对下一个单词进行处理。

5) 重复步骤 1)~步骤 4), 直到完成秘密信息的提取。

3 实验分析

本节对所提算法的性能进行评估实验, 并与现有主流生成式文本隐写算法进行整体性能和抗隐写分析性能的对比实验。

3.1 实验分析数据集

本文实验所有生成文本都是用参数量为 124 MB 的 GPT-2 预训练语言模型生成, 采用上文输入的生成方式, 上文信息选自 CNN/Daily Mail^[27, 28]数据集。数据集中的生成文本嵌入 200 bit 长度随机秘密消息。本文实验所使用的生成式隐写数据集的构成如表 3 所示。

Cover: 无任何秘密信息嵌入, 使用 124 MB 参数规模的 GPT-2 文本生成模型按照 top- $k(k=50)$ 的采样方式生成文本长度为 50 的 2 000 条样本, 作为 Cover 文本库, 是本文实验的对照组。

Stego-block: 基于定长编码的生成式文本隐写算法^[15]生成文本。在此算法中, 需要将生成式文本模型的候选池划分为 2^{block} 块, 同一块内的生成词嵌入的秘密信息相同, 参数 block 包含 3,4,5, 共 $2000 \times 3 = 6000$ 条文本。

Stego-Huffman: 基于哈夫曼编码的生成式文本隐写算法^[17]生成文本。使用变长编码的方式, 参数为候选池大小 m , m 包含 8, 16, 32 ($2^3, 2^4, 2^5$), 共 $2000 \times 3 = 6000$ 条文本。

Stego-Arithmetic: 基于算术编码的生成式文本隐写算法^[21]生成文本, 参数为 top- k 的 k 值, k 包含 300, 600, 900, 共 $2000 \times 3 = 6000$ 条文本。

表3 生成式隐写数据集的构成
Table 3 The composition of the generative steganographic text library

数据集名称	分组	数据集容量
cover	top- k ($k=50$)	2 000
stego-block ^[15]	block=3	2 000×3
	block=4	
	block=5	
stego-Huffman ^[17]	$m=8$	2 000×3
	$m=16$	
	$m=32$	
stego-Arithmetic ^[21]	top- k ($k=300$)	2 000×3
	top- k ($k=600$)	
	top- k ($k=900$)	
stego-SAAC ^[22]	$\delta=0.05$	2 000×3
	$\delta=0.10$	
	$\delta=0.15$	
stego-ppl	ppl _{thr} =30	2 000×3
	ppl _{thr} =60	
	ppl _{thr} =90	
stego-KL	kl _{thr} =0.1	2 000×3
	kl _{thr} =0.2	
	kl _{thr} =0.3	

Stego-SAAC: 基于算术编码的自适应生成式文本隐写算法^[22]生成文本。此算法的候选池截断策略为：根据语言模型输出的候选池概率分布，以候选池单词排序后累计概率值大于 $2^{-\delta}$ 选择相应 top- k 的 k 值， δ 包含 0.05, 0.1, 0.15 共 3 组，共 $2\,000 \times 3 = 6\,000$ 条。

Stego-ppl: 由本文所提出的基于困惑度可控截断产生的 k 值选择方案和算术隐写编码生成的文本，困惑度阈值包含 30, 60, 90 共 3 组，共 $2\,000 \times 3 = 6\,000$ 条。

Stego-KL: 由本文所提出的基于 KL 散度阈值的 k 值选择方案和算术隐写编码生成的文本，KL 散度阈值包含 0.1, 0.2, 0.3 共 3 组，共 $2\,000 \times 3 = 6\,000$ 条。

Stego-Arithmetic 和 **stego-SAAC** 是目前性能优异的生成式文本隐写算法，且它们的嵌入提取算法与本文 **stego-ppl** 和 **stego-KL** 相同，主要区别在于候选池截断策略不同，因此作为本文的对照组。

3.2 文本隐写分析实验设计

本文使用目前主流的文本隐写分析算法对正常生成文本和隐写生成文本进行训练检测来分析隐写算法的抗隐写分析能力。本实验采用的文本隐写分析方法包括 **FastText**^[29]、**TS-CNN**^[30]、**TS-RNN**^[31] 这 3 种方案。

FastText^[29] 是一个经典的文本分类算法，是 Facebook 在 2016 年开源的一个词向量计算和文本分类工具，**FastText** 的输入是多个单词及其额外的 n -gram 特征^[32]，隐含层是对多个词向量的叠加平均，输出时采用分层 **Softmax**。

TS-CNN^[30] 是一种基于语义分析的文本隐写分析方法，使用 **CNN** 提取文本的高层语义特征，利用秘密信息嵌入前后语义空间的细微分布差异。**TS-CNN**^[31] 卷积核的数量为 256，在卷积层后，需要使用最大池化的方式，同时需要 **Adam** 算法^[33] 训练 100 个 epoch，并且使用 **early stopping**^[34] 避免过拟合，检测阈值为 0.5。

TS-RNN^[31] 观察到隐写文本中每个单词的条件概率分布在嵌入秘密信息后会被扭曲，采用 **RNN** 提取这些特征分布差异，进行分析分类，同时利用这些细微差异来估计秘密信息数量。**TS-RNN** 使用双向 **GRU**^[35] 结构，利用 **Adam** 算法优化。使用 **tanh** 函数作为非线性激活函数。

本文实验所采用的训练和检测方法如下：首先对数据集进行筛选，去掉长度太短的句子，避免文本长度差距太大影响实验结果的准确性。筛选后的样本库需要根据 8:1:1 的比例划分训练集、验证集和测试集，并且截取每一句文本的前 30 词作为输入，词向量（word embedding）维度为 300。

3.3 性能评价指标

文本隐写算法的整体性能评价指标主要包括隐蔽性、统计安全性和嵌入容量。文本的隐蔽性通常使用困惑度进行衡量，困惑度越低，表明生成文本越自然，质量越好。统计安全性从 KL 散度和抗隐写分析能力两方面评价。KL 散度越低，说明文本隐写引入的特征差异越小，统计安全性越好；嵌入容量是生成文本中单词平均嵌入秘密信息的比特数。

抗隐写分析能力通过不同隐写分析方法检

测数据集的准确率 (Accuracy)、精确率 (Precision) 和召回率 (Recall) 来表示, 具体计算如式(8)~式(10)所示。

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (8)$$

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$

其中, TP (true positive) 是实际为隐写生成文本 stego, 检测分类也为 stego 的文本数量; TN (true negative) 是实际为正常生成文本 cover, 检测分类也为 cover 的文本数量; FP (false positive) 是实际为 cover, 被检测分类为 stego 的文本数量; FN (false negative) 是实际为 stego, 被检测分类为 cover 的文本数量。由于精确率和召回率通常呈现相反的趋势, 它们共同性能可以用 $F1$ 表示, 如式(11)所示。

$$F1 = \frac{2}{\frac{1}{\text{Pre}} + \frac{1}{\text{Recall}}} = \frac{2 \cdot \text{Pre} \cdot \text{Recall}}{\text{Pre} + \text{Recall}} \quad (11)$$

准确率 Acc 越小, 则表示生成隐写文本的抗隐写分析能力越强。

3.4 实验结果及分析

3.4.1 生成式文本隐写算法整体性能

本文算法 stego-ppl、stego-KL 和对比算法生成文本的困惑度、KL 散度、嵌入容量这 3 个方面的实验结果如表 4 所示。从表 4 中可以看出, 基于块状编码的生成式文本隐写算法^[15] (stego-block) 和基于哈夫曼编码的生成式文本隐写算法^[17] (stego-Huffman) 的生成样本在 KL 散度和嵌入容量上都不理想; stego-Arithmetic^[21] 和 stego-SAAC^[22] 的 3 个指标相较于前两种算法有了很大提升, 在困惑度相近的情况下, stego-SAAC 在 KL-散度和嵌入容量稍微优于 stego-Arithmetic, 幅度大约为 3%, 但在文献[22]的实验过程中仅仅比较了 KL 散度和嵌入容量两方面的内容, 根据本文实验的结果可以看出, 随着 δ 的减小, KL 散度与嵌入容量在性能上是同步提升的, 即 KL 散度减小, 嵌入容量增加, 因此, stego-SAAC 显示的性能会有较大幅度的提升。

表 4 各生成式文本隐写算法的整体性能分析
Table 4 The overall performance analysis of the above generation-based linguistic steganography

文本隐写算法	困惑度	KL 散度	嵌入容量
stego-block ^[15] (block=3)	49.59±9.99	2.62±0.21	3
stego-block ^[15] (block=4)	113.17±26.83	2.80±0.24	4
stego-block ^[15] (block=5)	252.18±69.04	2.93±0.27	5
stego-Huffman ^[17] (k=8)	11.46±1.91	1.08±0.14	2.41±0.14
stego-Huffman ^[17] (k=16)	14.52±2.95	0.85±0.14	2.98±0.19
stego-Huffman ^[17] (k=32)	18.31±4.24	0.68±0.14	3.48±0.27
stego-Arithmetic ^[21] (k=300)	22.49±8.17	0.23±0.09	4.15±0.51
stego-Arithmetic ^[21] (k=600)	27.284±11.22	0.20±0.09	4.42±0.57
stego-Arithmetic ^[21] (k=900)	30.07±13.24	0.18±0.09	4.58±0.61
stego-SAAC ^[22] ($\delta=0.05$)	34.28±18.33	0.16±0.08	4.74±0.72
stego-SAAC ^[22] ($\delta=0.10$)	27.93±14.03	0.19±0.08	4.44±0.67
stego-SAAC ^[22] ($\delta=0.15$)	24.63±11.61	0.21±0.08	4.25±0.64
stego-ppl _{thr} =30	23.52±4.15	0.22±0.16	4.31±0.24
stego-ppl _{thr} =60	35.63±12.63	0.13±0.10	4.89±0.51
stego-ppl _{thr} =90	39.99±17.46	0.08±0.07	5.06±0.64
stego-KL _{thr} =0.1	32.21±19.50	0.09±0.01	4.73±0.76
stego-KL _{thr} =0.2	22.79±10.94	0.18±0.01	4.15±0.66
stego-KL _{thr} =0.3	17.42±7.16	0.26±0.02	3.72±0.56

但是, 从表 4 可以看出, 这种性能上的提升是以困惑度这个指标为代价的, 当 KL 散度和嵌入容量都很优异时, 困惑度这个指标会不理想。当评价的指标变为 3 个 (困惑度、嵌入容量、KL 散度) 时, 本文提出的安全性可控的生成式文本隐写算法 stego-ppl、stego-KL 在 KL 散度和嵌入容量上表现最好, 与现有最新的 stego-SAAC 算法相比, 在 KL 散度相同时, 困惑度能下降 20%~30%, 同时嵌入容量不会受到影响。并且, 本文算法可以随时根据相关指标的阈值调整生成的文本, 使每个生成词的指标都能够满足阈值的要求, 相比其他方案更加直观和灵活。

表 5 是使用 stego-ppl 在 ppl_{thr}=60 的条件下的生成示例, 表中说明了生成每一个词时 KL 散度、困惑度、候选池截断的 k 值和嵌入比特数。可以看出, 虽然困惑度阈值为 60, 但是生成词的困惑度远远低于 60, 因此在设置阈值时可以偏大一点。

表5 本文困惑度可控的生成式文本隐写算法示例 ($ppl_{thr}=60$)Table 5 An example of generation-based linguistic steganography with controllable perplexity in this article ($ppl_{thr}=60$)

步骤 t	已生成的文本	k 值	困惑度	KL 散度	嵌入比特数	生成词
$t=1$	“”	9	15.22	0.59	2	“In”
$t=2$	“In”	27	10.9	0.21	2	“18”
$t=3$	“In 18”	73	28.79	0.01	9	“39”
$t=4$	“In 1839”	10	13.46	0.06	1	“,”
$t=5$	“In 1839,”	519	18.70	0.06	5	“President”
$t=6$	“In 1839, President”	433	17.95	0.03	4	“Abraham”
$t=7$	“In 1839, President Abraham”	53	11.89	0.001	0	“Lincoln”
$t=8$	“In 1839, President Abraham Lincoln”	5914	13.81	0.001	3	“declared”
$t=9$	“In 1839, President Abraham Lincoln declare”	9789	13.78	0.001	6	“war”
$t=10$	“In 1839, President Abraham Lincoln declare war”	3702	10.79	0.0002	1	“on”

在同时使用两种截断策略时，需要设置合理的阈值，使 ppl_{thr} 计算得到的 k 值大于 KL_{thr} 计算得到的 k 值，并将 ppl_{thr} 计算得到的 k 值作为最终确定的 k 值，因此适当偏大设置 ppl_{thr} 能够有更多的情况同时使用两种截断策略。

3.4.2 抗隐写分析能力

表6是本文算法 stego-ppl、stego-KL 和现有算法在不同文本隐写分析算法检测情况下的检测正确率 Acc 和 $F1$ ，体现了文本隐写算法的抗隐写分析能力。Acc 和 $F1$ 值越小，表示越不容易被检测，因此具有更好的统计安全性。从表6可以看出，stego-block^[15]的抗隐写分析能力最弱，不同参数下的检出率能够达到 80%~90%。其余算法具有良好的抗隐写分析性能，但是从检测准确率上来看，本文算法 stego-ppl、stego-KL 比 stego-Arithmetic 低 6%，比 stego-SAAC 低 3%，因此有着更优异的抗隐写分析能力。

Stego-Arithmetic^[21]和 stego-SAAC^[22]的生成样本在3种隐写分析算法检测下的 $F1$ 值优于 stego-Huffman^[17]生成样本。本文算法 stego-ppl、stego-KL 的生成文本在3种隐写分析算法检测下， $F1$ 值比 stego-SAAC^[22]和 stego-Arithmetic^[21]的生成文本平均值下降 0.050.1，部分情况下的 $F1$ 值都在 0.5 以下，具有更好的统计安全性。

表6 各生成式文本隐写算法的抗隐写分析能力分析
Table 6 Analysis of the anti-steganalysis capability of the above generation-based linguistic steganography

文本隐写算法	TS-CNN		TS-RNN		FastText	
	Acc	$F1$	Acc	$F1$	Acc	$F1$
stego-block ^[15] (block=3)	0.83	0.82	0.78	0.80	0.80	0.79
stego-block ^[15] (block=4)	0.86	0.87	0.88	0.86	0.88	0.87
stego-block ^[15] (block=5)	0.92	0.93	0.96	0.95	0.92	0.93
stego-Huffman ^[17] ($k=8$)	0.60	0.55	0.65	0.66	0.65	0.67
stego-Huffman ^[17] ($k=16$)	0.52	0.61	0.55	0.60	0.55	0.50
stego-Huffman ^[17] ($k=32$)	0.52	0.59	0.50	0.54	0.52	0.51
stego-Arithmetic ^[21] ($k=300$)	0.56	0.49	0.55	0.33	0.53	0.50
stego-Arithmetic ^[21] ($k=600$)	0.56	0.61	0.56	0.55	0.55	0.53
stego-Arithmetic ^[21] ($k=900$)	0.59	0.60	0.56	0.68	0.57	0.61
stego-SAAC ^[22] ($\delta=0.05$)	0.52	0.61	0.60	0.65	0.60	0.66
stego-SAAC ^[22] ($\delta=0.10$)	0.51	0.63	0.55	0.64	0.52	0.63
stego-SAAC ^[22] ($\delta=0.15$)	0.55	0.57	0.51	0.55	0.54	0.53
stego-ppl _{thr} =30	0.54	0.52	0.52	0.37	0.53	0.51
stego-ppl _{thr} =60	0.51	0.53	0.52	0.43	0.54	0.62
stego-ppl _{thr} =90	0.51	0.53	0.55	0.56	0.50	0.51
stego-KL _{thr} 0.1	0.54	0.42	0.53	0.54	0.53	0.36
stego-KL _{thr} =0.2	0.53	0.58	0.55	0.58	0.51	0.63
stego-KL _{thr} =0.3	0.50	0.51	0.58	0.66	0.53	0.52

4 结束语

针对现有生成式文本隐写算法在候选池的概率分布过于平坦或过于集中时, 生成的隐写文本在困惑度和 KL 散度方面存在不足的问题, 本文提出的安全性可控的生成式文本隐写算法, 基于 Transformer 的 GPT-2 文本生成模型和算术数编码的映射嵌入算法, 以困惑度和 KL 散度为参数限制, 生成隐写文本时设置困惑度或者 KL 散度的阈值, 使每个生成词都满足设定的指标阈值, 提高了生成文本的统计安全性。对比实验结果表明, 本文算法 stego-ppl、stego-KL 生成的隐写文本在困惑度、KL 散度和抗隐写分析性能上较现有算法有明显提升, 且生成文本的困惑度和 KL 散度可控。

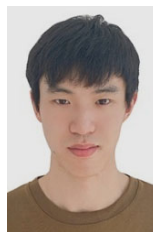
在相同 KL 散度, 嵌入容量不受影响的情况下, 本文算法 stego-ppl、stego-KL 生成文本的困惑度较现有算法 stego-SAAC^[22]和 stego-Arithmetic^[21]下降最高达 20%~30%, 生成文本在抗隐写分析性能上有部分提升, 3 种文本隐写分析方法检测的准确率均降低至 50%左右, 具有更好的抗隐写分析能力。

参考文献:

- [1] BENNETT K. Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text[R]. 2004.
- [2] XIANG L, WANG X, YANG C, et al. A novel linguistic steganography based on synonym run-length encoding[J]. IEICE Transactions on Information and Systems, 2017, 100(2): 313-322.
- [3] KHOSRAVI B, KHOSRAVI B, KHOSRAVI B, et al. A new method for pdf steganography in justified texts[J]. Journal of Information Security and Applications, 2019, 45(APR.):61-70.
- [4] ALATTAR A M, MEMON N D, HEITZENRATER C D, et al. Linguistic steganography on Twitter: hierarchical language modeling with manual interaction[C]//Media Watermarking, Security, & Forensics. International Society for Optics and Photonics. 2014: 902803.
- [5] DAI W, YU Y, DAI Y, et al. Text steganography system using markov chain source model and DES algorithm[J]. Journal of Software, 2010, 5(7): 785-792.
- [6] MORALDO H H. An Approach for text steganography based on Markov chains[J]. arXiv preprint arXiv: 1409.0915, 2014.
- [7] LUO Y, HUANG Y, LI F, et al. Text selenography based on cia-poetry generation using markov chain model[J]. Ksii Transactions on Internet & Information Systems, 2016, 10(9).
- [8] YANG Z, JIN S, HUANG Y, et al. Automatically generate steganographic text based on Markov model and Huffman coding[J]. arXiv preprint arXiv:1811.04720, 2018.
- [9] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
- [10] YANG Z L, ZHANG S Y, HU Y T, et al. VAE-Stega: linguistic steganography based on variational auto-encoder[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 880-895.
- [11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances In Neural Information Processing Systems. 2017: 5998-6008.
- [12] 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.
- [13] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.
- [14] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[J]. Advances in Neural Information Processing Systems, 2014, 27.
- [15] FANG T, JAGGI M, ARGYRAKI K. Generating steganographic text with LSTMs[J]. arXiv preprint arXiv:1705.10742, 2017.
- [16] YANG Z, ZHANG P, JIANG M, et al. Rits: real-time interactive text steganography based on automatic dialogue model[C]//International Conference on Cloud Computing and Security. 2018: 253-264.
- [17] 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, 2018, 14(5): 1280-1295.
- [18] HUFFMAN D A. A method for the construction of minimum-redundancy codes[J]. Proceedings of the IRE, 1952, 40(9): 1098-1101.
- [19] DAI F Z, CAI Z. Towards near-imperceptible steganographic text[J]. arXiv preprint arXiv: 1907.06679, 2019.
- [20] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Advances in Neural Information Processing Systems. 2014: 3104-3112.
- [21] ZIEGLER Z M, DENG Y, RUSH A M. Neural linguistic steganography[J]. arXiv preprint arXiv:1909.01496, 2019.
- [22] SHEN J, JI H, HAN J. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding[J]. arXiv preprint arXiv: 2010.00677, 2020.
- [23] WITTEN I H, NEAL R M, CLEARY J G. Arithmetic coding for data compression[J]. Communications of the ACM, 1987, 30(6): 520-540.
- [24] FAN A, LEWIS M, DAUPHIN Y. Hierarchical neural story generation[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018: 889-898.
- [25] HOLTZMAN A, BUYS J, FORBES M, et al. Learning to write with cooperative discriminators[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018: 1638-1649.
- [26] HOLTZMAN A, BUYS J, DU L, et al. The curious case of neural text degeneration[C]//International Conference on Learning Representations. 2019.
- [27] HERMANN K M, KOCISKY T, GREIFENSTETTE E, et al.

- Teaching machines to read and comprehend[J]. Advances in Neural Information Processing Systems, 2015, 28: 1693-1701.
- [28] NALLAPATI R, ZHOU B, GULCEHRE C, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond[J]. arXiv preprint arXiv:1602.06023, 2016.
- [29] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.
- [30] YANG Z, WEI N, SHENG J, et al. TS-CNN: Text steganalysis from semantic space based on convolutional neural network[J]. arXiv preprint arXiv:1810.08136, 2018.
- [31] YANG Z, WANG K, LI J, et al. TS-RNN: text steganalysis based on recurrent neural networks[J]. IEEE Signal Processing Letters, 2019, 26(12): 1743-1747.
- [32] WANG S I, MANNING C D. Baselines and bigrams: Simple, good sentiment and topic classification[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2012: 90-94.
- [33] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [34] PRECHELT L. Early stopping-but when[M]. Neural Networks: Tricks of the trade. Springer, Berlin, Heidelberg, 1998: 55-69.
- [35] ZHOU P, SHI W, TIAN J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016: 207-212.

[作者简介]



梅佳蒙（1997- ），男，湖北宜昌人，武汉大学硕士生，主要研究方向为信息隐藏。



任延珍（1973- ），女，陕西延安人，博士，武汉大学教授、博士生导师，主要研究方向为多媒体内容安全、AI 交互安全、多媒体信息隐藏和隐写分析。



王丽娜（1964- ），女，辽宁沈阳人，博士，武汉大学教授、博士生导师，主要研究方向为多媒体安全、云计算安全和网络安全。