Generative Text Steganography via Multiple Social Network Channels Based on Transformers

Long Yu[1,2], Yuliang Lu[1,2](✉), Xuehu Yan[1,2](✉), and Xianhui Wang[1,2]

[1] National University of Defense Technology, Hefei 230037, China
publicLuYL@126.com, publictiger@126.com
[2] Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei 230037, China

Abstract. Generative text steganography uses the conditional probability to encode the candidate words when generating tokens by language model, and then selects the corresponding word to output according to the secret message to be embedded, so as to generate stego text. The complex and open characteristics of social network provide a good camouflage environment for the transmission of stego texts, but also bring challenges: transmitting stego text through a single channel is easy to cause the destruction and loss of secret message; the speech of each social account needs to be combined with its background knowledge, so it has different language features. The existing text steganography schemes cannot solve these problems well. This paper proposes a multi-channel generative text steganography scheme in the context of social network, which hides secret message into multiple semantically natural texts, even if only a part of which can reconstruct secret message. Combined with the characteristics of social network, the bag-of-words models are used to control the topics of the stego texts in the process of text generation by language model. Two goal programming models are proposed to optimize the topic relevance and text quality of stego text. The experiment verifies the effectiveness of this scheme.

Keywords: Text steganography · Controllable text generation · Loss tolerance · Robustness · Imperceptibility

1 Introduction

With the wide development and application of the Internet and social network, digital information is easy to obtain, transmit and operate. Therefore, it is essential to protect sensitive information from malicious interference transmitting in public channels. Shannon [13] summarized three basic information security systems, namely, encryption system, privacy system and concealment system. The main purpose of encryption system is to protect the security of confidential message itself and privacy system aims to control access to confidential message.

W. Lu et al. (Eds.): NLPCC 2022, LNAI 13551, pp. 606–617, 2022.
https://doi.org/10.1007/978-3-031-17120-8_47

The concealment system hides confidential message into normal carriers and transmits them through open channels, paying attention to the protection of the existence of confidential message.

Steganography is a key technology of concealment system, which mainly studies how to embed secret information into carrier efficiently and safely. According to the different carrier types, steganography can be divided into image steganography [5], text steganography [7], audio steganography [10] and video steganography [8]. As the primary way of human communication from ancient times to the present, text has a wide range of application scenarios. And the transmission of text in the public channel is robust, because general channel doesn't compress it or interfere with it by noise. These show that texts may be more suitable as carriers for data transmission in social network than images, videos or other carriers.

Generative text steganography uses the language model (LM) to automatically generate stego text. It encodes the text semantic unit in the generation process, and selects the corresponding unit to output according to the secret message to be embedded, so as to realize the embedding of secret message. Therefore, the steganographer has greater freedom in the process of embedding message, so that a high information embedding rate can be expected. Yang et al. [17] proposed fix-length coding (FLC) based on perfect binary tree and variable-length coding (VLC) based on Huffman tree. They encode the Top-K words in the candidate pool predicted by the language model at each moment according to the conditional probability. Xiang et al. [16] modeled natural sentences as letter sequences and used the Char-RNN model to obtain letter-level conditional probability distributions. Zhou et al. [19] adopted an adversarial generative network model for steganographic text generation, and changed the construction method of candidate pool based on Top-K to dynamic candidate pool construction. However, the above schemes only consider the transmission of secret message through a single channel, and cannot effectively control semantic characteristics such as the topic of stego text.

The complex and open characteristics of social network provide a good camouflage environment for the transmission of stego texts, but also bring challenges. Since social networks are public channels, and each social platform is supervised by staff, if they find an account with abnormal behavior, it is likely to take measures to delete or ban the account. The transmission of stegotext through a single channel will result in the loss of secret message if the above situation is encountered. The (k, n) threshold secret sharing (SS) technology satisfies the characteristics of both encryption system and privacy system, which encrypts a secret message into n shares and distributes them. Any k shares can restore the original secret message, while less than k can obtain nothing. The loss-tolerant property of SS creates conditions for multi-channel transmission of secret message. Each social account has its own field of interest, professional direction and other backgrounds, thus possessing different language characteristics. If the semantics of the generated stego text can be effectively controlled in combination with the characteristics of social accounts, the concealment and security of covert communication through social network can be further improved. Controllable text generation (CTG) controls the characteristics of text, such as mood, style, etc., on the premise of ensuring the content [2,4,18]. CTG can model

$p(x|\alpha)$, where α is some expected controllable attribute, and x is the generated sample. Combining the characteristics of different social accounts to control the topics of each stego texts in the process of generation, the steganography scheme can be more suitable for application scenarios in the social network environment.

This paper proposes a multi-channel generative text steganography scheme with loss tolerance, robustness and imperceptibility in social network scenarios, which uses secret sharing technology to encrypt secret message into multiple shares, then the candidate words are encoded in the process of generation by a controlled language model, and the corresponding words output are selected according to the shares, so as to generate multiple topic-controlled stegotexts. We summarize the motivations and contributions of this paper as follows:

- Facing the challenge that the existing text steganography scheme only considers covert communication through a single channel, which can easily lead to the destruction or lost of stego text, this paper proposes to use the secret sharing technology to hide the secret message into multiple stego texts, and the original secret message can be recovered by only a part of them.
- In view of the characteristics of social network users' speech based on different backgrounds, this paper proposes to control the topics of the generated stego texts through bag of words (BoW), so that stego texts has stronger concealment.
- This paper proposes two goal programming models, which can optimize the topic relevance and text quality of stego text respectively.

2 Preliminaries and Related Work

2.1 Generative Text Steganography

In the field of natural language processing, text is usually regarded as a word sequence composed of specific words according to semantic association and syntactic rules, and the chain rule is used to describe the language model probability of the joint probability distribution of word sequences [1,9], whose expression is:

$$\begin{aligned} P(X) &= P(x_1, x_2, \ldots, x_N) \\ &= P(x_1)P(x_2|x_1)\cdots P(x_N|x_1x_2\cdots x_{N-1}) \\ &= \prod_{1}^{N} P(x_i|x_1x_2\cdots x_{i-1}) \end{aligned} \tag{1}$$

where $P(X)$ represents the generation probability of the word sequence x_1, x_2, $\cdots$,x_N, and $P(x_N|x_1x_2\cdots x_{N-1})$ denotes the conditional probability of generating word x_N given $x_1x_2\cdots x_{N-1}$ above. Due to the diversity of language expressions, for a given $x_1x_2\cdots x_{N-1}$, there will usually be more than one candidate x_N, which can make the generated text meet the constraints of semantic and syntactic rules. This provides redundancy for generative information hiding.

Yang et al. [17] proposed to use fixed length coding (FLC) based on a perfect binary tree with height h to encode the words in the candidate pool to achieve the mapping of secret bits to the word space. In the FLC scheme, the prefix text is input into LM to get the candidate words and their probability distribution for the next time step. Then, the candidate pool is truncated to 2^h in descending order of probability, and the candidate words are encoded by perfect binary tree, so that the corresponding words can be selected according to the secret bits to be embedded.

Perplexity (ppl) is usually used as the quality evaluation metric for generated text [6], as shown in Eq. 2.

$$\begin{aligned} ppl &= P(x_1, x_2, \cdots, x_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(x_i|x_1,x_2,\cdots,x_{i-1})}} \end{aligned} \tag{2}$$

from which we can see that the higher the conditional probability of the word sequence, the lower the perplexity, and the higher the quality.

2.2 Shamir's Polynomial-Based SS

Shamir's polynomial-based SS [12] for (k, n) threshold generates secret data m into n shares based on a $(k-1)$-degree polynomial as Eq. 3, in which $a_0 = m$, and $a_1, a_2, \cdots, a_{k-1}$ are assigned randomly in $[0, p-1]$ and p is a prime number greater than a_0. All modulo operations are performed in a galois field of $GF(p)$.

$$f(x) = (a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}) \bmod p \tag{3}$$

In the sharing phase, given n different random x, we can obtain n shared values by calculating $s_1 = f(x_1), s_2 = f(x_2), \cdots, s_n = f(x_n)$ and take (x_i, s_i) as a secret pair. These n pairs are distributed to n participants. Without loss of generality, x is often taken as $1, 2, \cdots, n$.

In the recovery phase, given any k pairs of $(x_i, s_i)|_{i=1}^{n}$, we can obtain the coefficients of $f(x)$ by Lagrange interpolation as shown in Eq. 4, and then $m = f(0)$.

$$f(x) = \sum_{j=1}^{k} f(i_j) \prod_{\substack{l=1 \\ l \neq j}}^{k} \frac{(x - i_l)}{(i_j - i_l)} \tag{4}$$

In this paper, we put l secret values into $a_i|_{i=0}^{l-1}$, and $a_i|_{i=l}^{k-1}$ are selected in $[0, p-1]$, which can effectively improve the efficiency of information hiding.

2.3 Transformer-Based Controllable Text Generation

Controllable text generation is based on the traditional text generation, adding the control of some attributes, styles, key information of the generated text, so that the generated text can meet our expectations.

Dathathri et al. proposed PPLM to [3] sample from the resulting $P(x|\alpha) \propto P(\alpha|x)P(x)$, and use a transformer [14] to model the distribution of natural language, thus effectively creates a conditional generative model. The following describes the principle of transformer and PPLM. The recurrent interpretation of a transformer [15] can be summarized as Eq. 5.

$$o_{t+1}, H_{t+1} = \mathrm{LM}(x_t, H_t) \tag{5}$$

where H_t is the history matrix consisting of key-value pairs from the past time-steps 0 to t. Then the x_{t+1} is sampled as $x_{t+1} \sim P_{t+1} = \mathrm{Softmax}(To_{t+1})$, where T is a linear transformation that maps the logit vector o_{t+1} to a vector of vocabulary size.

The probability distribution of words in the candidate pool at the next time step can be changed by adjusting H_t so that the probability of more relevant words to the topic is higher. Let ΔH_t be the update to H_t, generation with $(H_t + \Delta H_t)$ shifts the distribution of the generated text such that it is more likely to possess the desired attribute. ΔH_t is initialized at zero and PPLM rewrite the attribute model $P(\alpha|x)$ as $P(\alpha|H_t + \Delta H_t)$ and then make gradient based updates to ΔH_t as follows:

$$\Delta H_t \leftarrow \Delta H_t + \beta \frac{\nabla_{\Delta H_t} \log P(\alpha|H_t + \Delta H_t)}{\|\nabla_{\Delta H_t} \log P(\alpha|H_t + \Delta H_t)\|^{\gamma}} \tag{6}$$

where β is the step size, γ is the scaling coefficient for the normalization term. This update step can be repeated m times; in practice $m = 3$ to 10. Subsequently, a forward pass through the LM is performed to obtain the updated logits $\tilde{o}_{t+1}$ as $\tilde{o}_{t+1}, H_{t+1} = \mathrm{LM}(x_t, \tilde{H}_t)$, where $\tilde{H}_t = H_t + \Delta H_t$. The modified $\tilde{o}_{t+1}$ is then used to generate the new probability distribution $\tilde{P}_{t+1}$ at time step $t + 1$.

3 The Proposed Scheme

3.1 Information Hiding Algorithm

The schematic diagram of the hiding phase is shown in Fig. 1, where we take $h = 2$, $l = 1$ as an example, h is the height of perfect binary tree and l is the number of secret values to hide one time. We choose the smallest prime number greater than 2^h as p. First we slice secret bitstream in several units per h bits and convert these units into secret values in decimal integer form. Then we construct a $(k-1)-$degree polynomial as Eq. 3, and put l secret values in $a_0, a_1, \cdots, a_{l-1}$, the rest $k - l$ coefficients take values in the range $[0, p-1]$. Then the secret sharing module substitutes $x_i|_{i=1}^{n}$ into the polynomial to get n shared values $s_i|_{i=1}^{n}$. The mapping module uses the language model to continuously generate text, and modifies the probability distribution of each time step through BoW corresponding to a specific topic, so that the more topic compatible words in the candidate pool has the greater probability. Then perfect binary tree coding is carried out for the candidate words, corresponding words are selected according to the shared values and put into the stego text. All of the above processes are guided by the goal programming model (GPM).

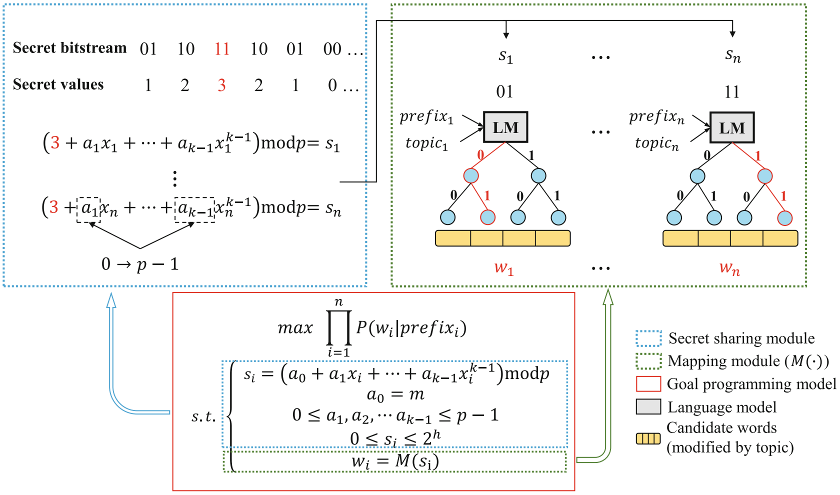
Secret bitstream 01 10 11 10 01 00 …
Secret values 1 2 3 2 1 0 …
$(3 + a_1x_1 + \cdots + a_{k-1}x_1^{k-1})\text{mod}p = s_1$
⋮
$(3 + a_1x_n + \cdots + a_{k-1}x_n^{k-1})\text{mod}p = s_n$
$0 \to p-1$
s_1
…
s_n
01
11
$prefix_1$
$topic_1$
LM
$prefix_n$
$topic_n$
LM
0
1
0
1
0
1
w_1
w_n
$max \prod_{i=1}^{n} P(w_i|prefix_i)$
$s.t.$
$s_i = (a_0 + a_1x_i + \cdots + a_{k-1}x_i^{k-1})\text{mod}p$
$a_0 = m$
$0 \leq a_1, a_2, \cdots a_{k-1} \leq p-1$
$0 \leq s_i \leq 2^h$
$w_i = M(s_i)$
Secret sharing module
Mapping module ($M(\cdot)$)
Goal programming model
Language model
Candidate words (modified by topic)

Fig. 1. The schematic diagram of the hiding phase.

The attribute model used in this scheme is the BoWs corresponding to different topics. A BoW is a set of keywords $\{word_1, \cdots, word_z\}$ that specify a topic. $\log P(\alpha|x)$ can be represented as Eq. 7.

$$\log P(\alpha|x) = \log(\sum_{i}^{z} P_{t+1}[word_i]) \tag{7}$$

where P_{t+1} is the conditional probability distribution of the output of the language model at moment $t+1$. We can calculate ΔH_t by Eq. 6 to modify H_t and finally obtain the conditional probability distribution $\tilde{P}_{t+1}$ that satisfies the particular topic.

We propose two goal programming models (GPM-topic and GPM-ppl) to optimize the topic relevance and text quality of the generated stego texts for different applications, respectively. GPM-topic is expressed as Eq. 8.

$$\max \prod_{i=1}^{n} \tilde{P}(w_i|prefix_i)$$
$$s.t. \begin{cases} s_i = (a_0 + a_1x_i + \cdots + a_{k-1}{x_i}^{k-1}) \bmod p \\ a_i = m_i|_{i=0}^{l-1} \\ 0 \leq a_l, a_{l+1}, \cdots, a_{k-1} \leq p-1 \\ 0 \leq s_i \leq 2^h \\ w_i = M(s_i) \end{cases} \tag{8}$$

where $\tilde{P}(w_i|prefix_i)$ represents the conditional probability of generating the next word w_i when the prior words $prefix_i$ of the i-th stego text is determined, $\tilde{P}$ is modified by BoW_i to make the word probability more relevant to $topic_i$, and

Algorithm 1. The information hiding phase of the proposed scheme.

```
Input: Secret bitstream $B$; $(k, n)$threshold; the prime number $p$; $x_1, \cdots, x_n$; height
    of perfect binary tree $h$; the number of secret units to hide at one time $l$; the topics
    of each stego text $topic_1, \cdots, topic_n$; bag of words $BoW_1, \cdots, BoW_n$ related to
    $topic_i$; initial words for each stego text $prefix_1, \cdots, prefix_n$ (is what the LM is
    conditioned on to generate a passage of text); language model LM.
Output: $n$ stego texts $ST_1, \cdots, ST_n$.
$B$ is sliced per $h$ bits, and each unit is converted to integer form to obtain the
    sequence of secret values;
for each $prefix_i$ do
    Input $prefix_i$ into LM to get the initial history state ${H_t}^{\mathrm{i}}$ of $ST_i$;
    $ST_i \leftarrow prefix_i$;
while not the end of the sequence of secret values do
    if not achieve the goal of GPM then
        Construct a polynomial as Eq. 3, whose first $l$ coefficients are consecutive $l$
            secret values and the rest $k - l$ coefficients are chosen from $[0, p - 1]$;
        Put $x_1, \cdots, x_n$ into polynomial to get $n$ shared values $s_1, \cdots, s_n$;
        for each $s_i$ do
            According to $BoW_i$, using Eq. 6 and Eq. 7 to obtain $\Delta {H_t}^i$, then we can get
                the history status $\tilde{H}_t^i \leftarrow {H_t}^i + \Delta {H_t}^i$ at this moment modified by $topic_i$;
            Input $\tilde{H}_t^i$ and the last word of $ST_i$ into LM to get the modified logits $\tilde{o}_{t+1}$,
                and softmax $\tilde{o}_{t+1}$ to get the conditional probability distribution $\tilde{P}_{t+1}$ that fits the
                $topic_i$ at time $t + 1$, then arrange $\tilde{P}_{t+1}$ in descending order, and take the first $2^h$
                words to form the candidate pool;
            The words in the candidate pool are encoded by a perfect binary tree, and
                the corresponding word $w_i$ is selected based on the shared value $s_i$;
    else
        Add $w_i$ to $ST_i$;
return $ST_1, \cdots, ST_n$
```

$m_i|_{i=0}^{l-1}$ are the consecutive l secret values. $M(\cdot)$ represents the mapping module that maps the shared value s_i through perfect binary tree encoding to the LM-generated word space. Since we choose to put the secret values in the first l coefficients of Eq. 3, the remaining $k - l$ elements are selected from $[0, p - 1]$, which makes the shared values not unique for the same set of secret values. So we can get different combinations of words to output by constantly adjusting the last $k - l$ coefficients of the polynomial. The goal in GPM-topic is to take advantage of this to find the word combination with the largest conditional probability product, i.e., the combination with the strongest relevance to their respective topics, in order to generate more appropriate stego texts. Since the size of the candidate pool is smaller than p, and the operations of SS are all under $GF(p)$, the value range of s_i is $[0, p - 1]$ if no control is applied, so the selection of words will be out of the range of the candidate pool. Therefore, we limit the value of s_i in the constraints of GPM, which can be also achieved by adjusting the $k - l$ coefficients of the polynomial.

Algorithm 2. The information extraction phase of the proposed scheme.

```
Input: $k$ stego texts $ST_1, \cdots, ST_k$; $x_1, \cdots, x_k$; height of perfect binary tree $h$; the number of secret units to hide at one time $l$; the topics of each stego text $topic_1, \cdots, topic_k$; bag of words $BoW_1, \cdots, BoW_k$ related to $topic_i$; language model LM.
Output: Original secret bitstream $B$.
for each stego text $ST_i$ do
    Input the prefix in $ST_i$ into LM to get the original initial history state $H_t$;
    while not the end of $ST_i$ do
        According to $BoW_i$, using Eq. 6 and Eq. 7 to obtain $\Delta H_t$, then we can get the history status $\tilde{H}_t \leftarrow H_t + \Delta H_t$ at this moment modified by $topic_i$;
        Input $\tilde{H}_t$ and the last word of $ST_i$ into LM to get the modified logits $\tilde{o}_{t+1}$, and softmax $\tilde{o}_{t+1}$ to get the conditional probability distribution $\tilde{P}_{t+1}$ that fits the $topic_i$ at time $t+1$, then arrange $\tilde{P}_{t+1}$ in descending order, and take the first $2^h$ words to form the candidate pool;
        Use a perfect binary tree to encode the words in the candidate pool, the codeword corresponding to $x_{t+1}$ is extracted and converted into integer form, then the shared value $s_i$ is obtained, which is then added to $Shares_i$;
for each $s_i$ in each $Shares_i$ do
    Put $k$ pairs $(x_i, s_i)|_{i=1}^{k}$ into Eq. 4, then we can recover a $(k-1)-$degree polynomial, whose first $l$ coefficients are the consecutive $l$ secret values, add them to the secret value sequence;
Each integer in the sequence of secret values is converted into the binary form of $h$ bits, then the original secret bitstream $B$ is obtained;
return $B$
```

In the mapping module, we modify the original probability distribution P_{t+1} by using BoW to obtain $\tilde{P}_{t+1}$ with a higher probability of fitting the topic. However, the language model uses a large amount of natural texts for training to fit the natural language distribution, and modifying it will affect the quality of the generated text, which is the cost of enhancing the relevance of the text topic. Inspired by Eq. 2, we propose GPM-ppl to improve the quality of stego text. The form of GPM-ppl is consistent with Eq. 8, except that the modified probability $\tilde{P}$ in the goal is replaced by the original probability distribution P obtained by LM. Therefore, we can find the word combination with the largest original probability product while satisfying the constraints, so that each word and its previous words are closer to the original distribution, thus reducing the perplexity and improving the quality of stego text. But at the same time, this reduces the likelihood of selecting words that match the topic, which inevitably reduces the topic relevance of stego text. Therefore, the choice of GPM should be determined according to the requirements of actual application scenarios.

Algorithm details of the proposed hiding method are shown in Algorithm 1.

3.2 Information Extraction Algorithm

When k or more stego texts are obtained, the extraction of secret message can be performed. The inverse mapping module generates the conditional probability distribution of the next word through the same text generation method as the hiding phase and encodes the candidate pool using a perfect binary tree. Because stego texts are deterministic, there is no need to select candidate words similar to the sampling strategy in the hiding phase, but to find the corresponding codewords to get the shared values. After that, the reconstruct module can recover a polynomial with the shared values using Eq. 4, whose first l coefficients are secret values. Algorithm 2 shows the detailed process of extraction. For the convenience of representation and without loss of generality, we assume that the k stego texts obtained are the first k of the n stego texts.

4 Experiments and Ablation Study

4.1 Experimental Setup

We evaluate the performance of the proposed scheme on a public corpora "A Million News Headlines", which contains 1,226,259 sentences on news headlines published by the Australian news source ABC (Australian Broadcasting Corporation) over an eighteen-year period. We randomly select 100 sentences from the dataset for experiments. We use the 345M parameter GPT-2 model [11] based on the transformer architecture as the text generation model.

To evaluate the quality of stego text we use the perplexity as Eq. 2. For topic relevance, there is no good evaluation index in the current study. Since the purpose of topic control is achieved by BoW adjusting the conditional probability distribution, we decide to use the percentage of words in the stego text belonging to BoW_i to evaluate the topic relevance (TR) with $topic_i$, as shown in Eq. 9.

$$TR_i = \frac{N_{BOW_i}}{N} \times 100\% \tag{9}$$

where TR_i represents the topic relevance of ST_i related to $topic_i$, N is the number of words in ST_i, and N_{BOW_i} represents the number of words in ST_i that appear in BoW_i.

4.2 Effectiveness Demonstration

The hyperparameters of the proposed scheme include (k, n) threshold, the prime number p; the number of secret values to hide at one time l, the topic of each stego text $topic_i$, the height of the perfect binary tree h, and the initial words of each stego text $prefix_i$. Below we show the actual effect of the proposed scheme when these parameters are taken at different values, as shown in Tables 1 and 2. We choose "Secret message" as the secret text. The target topics of stego texts are colored and bracketed (e.g. **[military]**). The words that appear in BoW are highlighted brightly (e.g., tank). Softer highlighting corresponds to words related to the topic but not in BoW (e.g., turret). The prefix of each sentence is underlined (e.g., More importantly).

Table 1. Stego texts of "Secret message" when $k = 2$, $n = 3$, $l = 1$, $h = 3$, $p = 11$.

ST_1 **[military]**	More importantly though I can now see what the problem will do to me and I am not a tank and am not getting damage done so far. This will probably cause the enemy team turret tanks tank to get hit and killed
ST_2 **[science]**	The connection is that we have all become part-time scientists at some of our own research institutions we have our own experiments running, we have a team in residence lab working under contract at another institute laboring in the
ST_3 **[legal]**	It has been shown in several articles that people do indeed believe the truth when presented a compelling case for why an issue merits a ban for both criminal and national defence laws to include an issue as evidence of their legality and for a

Table 2. Stego texts of "Secret message" when $k = 3$, $n = 4$, $l = 2$, $h = 3$, $p = 11$.

ST_1 **[technology]**	In brief overview: We're building out new API end point to help with web services in Java 9 (Java 10
ST_2 **[politics]**	The key aspect of all the arguments that are raised against a state's constitutional power of legislative self governance the authority over
ST_3 **[religion]**	It has been shown time, that there can always and surely follow in nature a Divine God and God-Man. And God
ST_4 **[space]**	To review some more details about a project like Spacecraft Launch Mission we'll have some of these satellites orbit our moon

4.3 Ablation Study

We conduct an ablation study with five variants: **B**: the baseline, no topic control, no GPM (that is, the conditional probability distribution is not modified using BoW, and $a_i|_{i=l}^{k-1}$ are chosen randomly); **BP**: no topic control, GPM-ppl; **BT**: topic control, no GPM; **BTP**: topic control, GPM-ppl; **BTT**: topic control, GPM-topic.

We use the 100 sentences selected from Sect. 4.1 as the secret texts and hide them using each of the above five methods, and count the average perplexity and topic relevance of each stego text. The experimental results are shown in Tables 3 and 4.

Through the above experimental results we can draw the following conclusions.

Table 3. Average ppl and TR of stego texts when $k = 2$, $n = 3$, $l = 1$, $h = 3$, $p = 11$

Variants	B	BP	BT	BTP	BTT
Avg. ppl ↓	32.89	13.88	42.21	16.17	18.37
Avg. TR ↑	\	\	7.56 %	4.44 %	13.42 %

Table 4. Average ppl and TR of stego texts when $k = 3$, $n = 4$, $l = 2$, $h = 3$, $p = 11$

Variants	B	BP	BT	BTP	BTT
Avg. ppl ↓	36.00	20.65	53.93	28.46	32.41
Avg. TR ↑	\	\	10.9 %	5.56 %	11.55 %

- In this scheme, the topic control method can effectively increase the probability of the words matching the topic being selected in the process of stego text generation, so that the stego text can meet the specific topic.
- The text quality is affected because the topic control method modifies the probability distribution in the process of text generation, which makes the modified probability distribution inconsistent with the training sample. Therefore, the text quality of the BT method without the optimization of GPM is the worst.
- The BP method optimized by GPM-ppl generates the highest quality stego text, and the perplexity of GPM-ppl optimized BTP method is less than that of BT and BTT, so GPM-ppl can effectively improve the quality of stego text.
- The topic relevance of the BTT method optimized by GPM-topic is the highest, so GPM-topic can effectively improve the topic relevance of stego text.

5 Conclusions

In this paper, we propose a text steganography scheme with loss tolerance, robustness, and imperceptibility, which hides secret message into n fluent and topic-controlled stego texts, where any k or more stego texts can recover the secret message. We first use secret sharing to encrypt secret message into shared values. Then, we use bag-of-words model to modify the conditional probability distribution to make the probability of words that fit the topic larger. Finally, a perfect binary tree is used to map shared values to the word space to generate stego texts. We also propose two goal programming models to optimize topic relevance and text quality of stego texts respectively. In the experimental section, we show some practical examples and perform ablation experiments to illustrate the effectiveness of each module.

References

1. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. In: Advances in Neural Information Processing Systems 13 (2000)

2. Chan, A., Ong, Y.S., Pung, B., Zhang, A., Fu, J.: CoCon: a self-supervised approach for controlled text generation. In: International Conference on Learning Representations (2020)
3. Dathathri, S., et al.: Plug and play language models: a simple approach to controlled text generation. In: International Conference on Learning Representations (2019)
4. Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Toward controlled generation of text. In: International conference on machine learning, pp. 1587–1596. PMLR (2017)
5. Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T., Jung, K.H.: Image steganography in spatial domain: a survey. Sig. Process. Image Commun. **65**, 46–66 (2018)
6. Jurafsky, D.: Speech & language processing. Pearson Education India (2000)
7. Krishnan, R.B., Thandra, P.K., Baba, M.S.: An overview of text steganography. In: 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–6. IEEE (2017)
8. Liu, Y., Liu, S., Wang, Y., Zhao, H., Liu, S.: Video steganography: a review. Neurocomputing **335**, 238–250 (2019)
9. Manning, C., Schutze, H.: Foundations of statistical natural language processing. MIT press (1999)
10. Mishra, S., Yadav, V.K., Trivedi, M.C., Shrimali, T.: Audio steganography techniques: a survey. In: Bhatia, S.K., Mishra, K.K., Tiwari, S., Singh, V.K. (eds.) Advances in Computer and Computational Sciences. AISC, vol. 554, pp. 581–589. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-3773-3_56
11. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
12. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
13. Shannon, C.E.: Communication theory of secrecy systems. Bell Syst. Tech. J. **28**(4), 656–715 (1949)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
15. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)
16. Xiang, L., Yang, S., Liu, Y., Li, Q., Zhu, C.: Novel linguistic steganography based on character-level text generation. Mathematics **8**(9), 1558 (2020)
17. Yang, Z.L., Guo, X.Q., Chen, Z.M., Huang, Y.F., Zhang, Y.J.: Rnn-stega: linguistic steganography based on recurrent neural networks. IEEE Trans. Inf. Forensics Secur. **14**(5), 1280–1295 (2018)
18. Zellers, R., et al.: Defending against neural fake news. In: Advances in Neural Information Processing Systems 32 (2019)
19. Zhou, X., Peng, W., Yang, B., Wen, J., Xue, Y., Zhong, P.: Linguistic steganography based on adaptive probability distribution. In: IEEE Transactions on Dependable and Secure Computing (2021)