Topic-aware Neural Linguistic Steganography Based on Knowledge Graphs

YAMIN LI and JUN ZHANG, Hubei University, China
ZHONGLIANG YANG, Tsinghua University, China
RU ZHANG, Beijing University of Posts and Telecommunications, China

The core challenge of steganography is always how to improve the hidden capacity and the concealment. Most current generation-based linguistic steganography methods only consider the probability distribution between text characters, and the emotion and topic of the generated steganographic text are uncontrollable. Especially for long texts, generating several sentences related to a topic and displaying overall coherence and discourse-relatedness can ensure better concealment. In this article, we address the problem of generating coherent multi-sentence texts for better concealment, and a topic-aware neural linguistic steganography method that can generate a steganographic paragraph with a specific topic is present. We achieve a topic-controllable steganographic long text generation by encoding the related entities and their relationships from Knowledge Graphs. Experimental results illustrate that the proposed method can guarantee both the quality of the generated steganographic text and its relevance to a specific topic. The proposed model can be widely used in covert communication, privacy protection, and many other areas of information security.

CCS Concepts: • **Security and privacy** → **Privacy protections**; *Cryptography*;

Additional Key Words and Phrases: Neural networks, linguistic steganography, knowledge graph, topic aware, text generation

ACM Reference format:
Yamin Li, Jun Zhang, Zhongliang Yang, and Ru Zhang. 2021. Topic-aware Neural Linguistic Steganography Based on Knowledge Graphs. *ACM/IMS Trans. Data Sci.* 2, 2, Article 10 (March 2021), 13 pages.
https://doi.org/10.1145/3418598

1 INTRODUCTION

As a new stage in the development of information technology, big data has become a topic of great concern. The trend of big data and digitation promotes the development of information security. Information hiding (also known as steganography), one of the key technologies of information security [24], has attracted wide attention of researchers in recent years [7, 11, 17, 35–37].

This research is supported by the National Natural Science Foundation of China under Grant No. U1936216 and the Natural Science Foundation of Hubei Province under Grant No. 2018CFC900.

Authors' addresses: Y. Li and J. Zhang, Hubei University, 368 Youyi Avenue, Wuchang District, Wuhan, Hubei Province, China, 430062; emails: yamin.li@hubu.edu.cn, 201911111911542@stu.hubu.edu.cn; Z. Yang, Tsinghua University, 30 Shuangqing Road, Haidian Qu, Beijing Shi, China, 100091; email: yangzl15@mails.tsinghua.edu.cn; R. Zhang, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Haidian Qu, Beijing Shi, China, 100876; email: zhangru@bupt.edu.cn.

2577-3224/2021/03-ART10 $15.00
https://doi.org/10.1145/3418598

Compared with traditional cryptography, which encrypts a secret message into an unintelligible form, steganography encodes a secret message into common information carriers to conceals that communication is taking place. That is, the security of steganography derives from the inability to detect that a message exists within the cover signal [31]. Therefore, steganography is widely used in military intelligence support, covert communication, privacy protection, copyright protection, and other fields [8].

Nowadays, various digital media such as images [36], audio [34, 37], and text [7, 11, 17, 19, 27, 28, 35] have become popular carriers for information transmission, as well as information hiding. Linguistic steganography, which uses text as the information hiding carrier, has appealed to a tremendous proportion of researchers' interests, since natural language is a widely used information carrier in communication [30]. However, compared with image or audio, text has a higher degree of information coding, resulting in less redundant information, which makes it quite challenging to hide secret information [8, 38]. Therefore, the researches of linguistic steganography have great research value and practical significance. More and more methods and implementations have emerged.

Linguistic steganography methods can be classified as selection-based, modification-based, and generation-based [8]. Selection-based linguistic steganography simply selects different text cover to represent different meanings [12, 13]. Modification-based methods are more popular, such as text format modification [5] and synonym substitution [15, 16]. These methods are obviously with a low security and embedding rate, and are difficult to put into practice. In generation-based method, an entire block of text is generated while encoding the message reversibly in the choice of tokens [28, 38, 39]. Thus, it can have a higher hidden capacity and has become a promising research direction in the field.

Current researches of generation-based linguistic steganography mainly use the framework consists of two modules: text generation model and probability distribution-based encoding method [35, 38]. First, they use a well-designed model to learn a statistical distribution model of a large number of natural texts. Then encoded the conditional probability distribution of each word in the text generation process according to the secret information. This linguistic steganography framework faces great challenges. First, as the embedding rate increases, the quality of the generated text decreases, and even meaningless sentences with grammatical errors appear. How to design a better language generation model and generate higher quality text carriers with smoother sentences is a problem that needs to be solved. Second, the linguistic steganography methods based on the above framework only generate text carriers according to the statistical distribution of probability. The semantics, emotions and topics of the generated text are uncontrollable. To improve the concealment of the generated text, in addition to increasing sentence fluency, we also expect that the generated text has a certain topic and consistent emotion, just like the natural language used by human beings. How to generate natural language with controllable topics and emotions has become an urgent issue.

Thanks to the development of machine learning, neural network-based steganography approaches with better language generation model have begun to appear and being studied [38, 39]. It can solve the first challenge to a certain extent. However, the contradiction between embedding rate and text quality has always existed, and the emotion and topic of the generated text are uncontrollable, which greatly affects the concealment and practicality of the steganography method. When generating text, we must not only consider the probability distribution between text characters but also consider the problem at a higher level, such as semantics, emotions, and topics. Therefore, in this article, a topic-aware neural linguistic steganography method is present, which can generate a steganographic paragraph with a specific topic. We achieve it by Introducing data from Knowledge Graph (KG) to guide the generation produce. The great benefit of introducing knowledge graph is that knowledge graph provides data of relevant topic and content, since there must be a relationship between the entities in a knowledge graph. This ensures that the text generated by the model has associated semantics and can support the certain topic.

2 RELATED WORKS

Generation-based linguistic steganography has become a hot research topic in the field of information security because of its high concealment and practicality. Many works have emerged in recent years [18, 25, 28, 30, 35, 38, 39]. As we mentioned above, the current generation-based linguistic steganography methods mainly combine with language generation models based on statistical distribution of probability. They use the framework consists of two modules: text generation model and probability distribution-based encoding method.

Some early researches use the Markov chain model [18, 25, 30, 35] to calculate the common occurrences of each phrase and obtain the transition probability, then they encode the words using the transition probability to achieve the purpose of hiding secret information in the text generation process. However, due to the limitations of the Markov model, the quality of generated text is limited and it is easily detected by the text steganalysis algorithm.

With the widespread application of neural networks in the field of neural language processing, some researchers have proposed linguistic steganography methods based on language generation models using neural networks. Fang et al. [28] design a linguistic stegosystem based on a Long Short-Term Memory (LSTM) neural network. They use the LSTM network to learn the statistical language model from natural text. And in the text generation process, choose different words from a pre-encoded dictionary according to the secret information. Compared with the previous Markov-based methods, the system has the advantage of encoding much more information.

Yang et al. [38] also use a multi-layer Recurrent Neural Networks (RNN) neural network with LSTM unit to train a statistical language model from a large number of natural language samples, and calculate the conditional probability distribution of each element. Then they proposed two different encoding methods, which are fixed-length coding and Huffman coding to encode the conditional probability distribution of each word and output corresponding words according to the secret information.

Afterwards, Ziegler et al. [39] further improved the text generation model and the coding method of conditional probability distribution. They propose a linguistic steganography approach combining arithmetic coding with 345M parameter GPT-2 model [23], which is one of the state-of-the-art language model. The pre-trained language model GPT-2, trained with a large and diverse dataset, can generate text that is more close to the probability distribution of the neural language. In addition, they have proved arithmetic coding is more effective than Huffman coding and has less damage to the probability distribution.

Benefit from better neural network-based text generation models and more efficient coding methods, the quality of generated steganographic texts have been significantly improved. However, in practical applications, such as covert communication, a sufficiently natural text is not absolutely secure. The content and topic of the text need to conform to a specific context. Especially for long paragraphs with multiple sentences, the text must be consistent with the context and topic. A recent research by Yang et al. [32] reveals that even if the quality of the generated text is good enough, there are still potential security risks due to the uncontrollable semantics.

At present, the above-mentioned linguistic steganography methods only generate text containing secret message using statistical distributions of probability. When generating, the emotion and topic of the generated text are uncontrollable, which greatly affects the concealment and practicality of the steganography method. Of course, if we use the corpus related to a certain topic to train the generation model, then the generated text can be related to a specific topic, but the applicability of this method will be very limited.

Therefore, in this article, a topic-aware neural linguistic steganography method is present, which can generate a steganographic paragraph with a specific topic. By Introducing entities and relationships data from Knowledge Graph, the text generated by the model has the same associated semantics and topic as the knowledge graph.

3 PROBLEM FORMULATION

Generally, a steganography task can be expressed as “The Prisoners’ Problem” [26]: Alice and Bob are separated in prison, and they need to pass some secret information without being discovered by the warden Eve. So Alice and Bob attempt to hide the secret information in the carrier. Suppose that Alice and Bob need to transmit some secret message m in the secret message space $\mathcal{M}$. Alice gets a cover c from the cover space $\mathcal{C}$. Under the guidance of a certain key k_A in the keys space $\mathcal{K}$, the mapping function f is used to map c to s, which is in the secret hidden space $\mathcal{S}$, that is

$$Emb : \mathcal{M} \times \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{S}, \quad f(m, c, k_A) = s. \tag{1}$$

Bob uses the extraction function g to extract the correct secret message m from the hidden object s under the guidance of the key k_B in the keys space $\mathcal{K}$:

$$Ext : \mathcal{S} \times \mathcal{K} \rightarrow \mathcal{M}, \quad g(s, k_B) = m. \tag{2}$$

To prevent the transmission of information from being suspected by Eve, the steganography operation should not cause a large difference in the distribution of the carriers in the semantic space. That is, the probability distributions of $\mathcal{S}$ and $\mathcal{C}$ should be as close as possible:

$$d_f(P_\mathcal{C}, P_\mathcal{S}) \leq \epsilon. \tag{3}$$

In this article, the task is to design a topic-aware neural linguistic steganography model that can introduce data from knowledge graph and generate steganographic text with a specific topic. We formulate our problem as follows: given a topic-specific graph g in the knowledge graph space $\mathcal{G}$ and the secret message m, the goal is to generate a steganographic text s that (a) is a smooth natural language paragraph, (b) expresses the certain topic of the knowledge graph, and (c) hides secret message and can be extracted. And in this generation-based steganography method, the cover c is no longer needed, since the steganographic text s is automatically generated based on confidential information. Therefore, the embedding function Equation (1) can be rewritten as

$$Emb : \mathcal{M} \times \mathcal{K} \times \mathcal{G} \rightarrow \mathcal{S}, \quad f(m, k_A, g) = s. \tag{4}$$

And it still has to satisfy Equation (3), that is, the steganographic operation should minimize the impact of the carrier on the semantic spatial distribution.

4 THE PROPOSED MODEL

The proposed topic-aware neural linguistic steganography model also follows the Encoder-Decoder architecture. We try to introduce a topic-specific graph at the encoder side, use a graph encoding network to embed the topic and content of the graph into vectors. Then send it to the decoder to generate steganographic text.

The overall architecture is shown in Figure 1. The input of the proposed model is the secret message bitstream and a topic-specific graph that is encoded with a graph embedding network based on Transformer architecture, which is discussed in Section 4.1. At each decoder time step, we attend on encodings of the knowledge graph using the decoder hidden state. The resulting vectors are used to select the output either from the decoder’s vocabulary or by copying an entity from the knowledge graph. Details of our decoding process are described in Section 4.2.

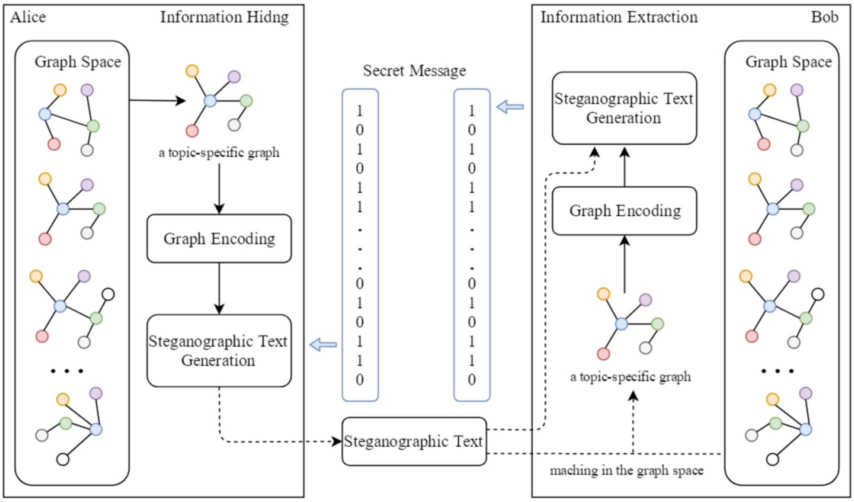

Fig. 1. The overall architecture of the proposed method.

4.1 Graph Encoding

To introduce the topic and content of the graph, we design a graph embedding network to extract the semantics of the entities and relationships in the graph and convert them into corresponding graph vectors. The network is mainly refer to the method proposed in Reference [14] that based on the Transformer architecture [29]. The Transformer architecture addresses the inherent sequential computation shortcoming of recurrent neural networks, enabling efficient and paralleled computation by invoking a self-attention mechanism for global context modeling. The models based on Transformer architecture have been used in varieties of natural language processing applications and proved to be effective [4, 22].

First, we convert each topic-specific graph g in the graph space $\mathcal{G}$ to an unlabeled connected bipartite graph g'. A graph consists of vertices and edges. We define a topic-specific graph $g = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of edges. And the converted graph g' is defined as $g' = \{\mathcal{V}', \mathcal{E}'\}$. The new vertices set $\mathcal{V}'$ contains the original vertices $\mathcal{V}$ and the vertices transformed from the original edges $\mathcal{E}$. Each edge is transformed into two vertices: one representing the forward direction of the relation and one representing the reverse. These new vertices are then connected to the original vertices to form the new edges set $\mathcal{E}'$, which is actually an adjacency matrix describing the directed edges. After the conversion, the directionality of the former edge is maintained and there is no loss of the graph information. More importantly, it can effectively avoid parameter explosion cased by the labelled edges in the graph-based encoders [2].

Second, we embedding the new vertices $\mathcal{V}'$ into a dense continuous space matrix. Entities correspond to knowledge graph terms that are often multi-word expressions. We use the last hidden state of a bidirectional RNN run over embeddings of each word in the entity phrase to produce a single d-dimensional embedding per phrase. The output of the vertices embedding step is matrix $V^0 = [\boldsymbol{v}_i]$, $\boldsymbol{v}_i \in \mathbb{R}^d$ representing each vertex in $\mathcal{V}'$.

Third, a multi-head self attention mechanism is used [29]. Each vertex representation $\boldsymbol{v}_i$ is contextualized by attending his neighbours in g':

$$a_{ij}^n = \frac{\exp(a^n(\boldsymbol{v}_i, \boldsymbol{v}_j))}{\sum_{j \in \mathcal{N}_i} \exp(a^n(\boldsymbol{v}_i, \boldsymbol{v}_j))}, \tag{5}$$

$$\hat{\boldsymbol{v}}_{\boldsymbol{i}} = \boldsymbol{v}_i + \|_{n=1}^{N} \sum_{j \in \mathcal{N}_i} a_{ij}^n \boldsymbol{W}^n \boldsymbol{v}_j, \tag{6}$$

where N is the number of attention heads, $\|$ denotes the concatenation of the N attention heads, $\mathcal{N}_i$ denotes the neighborhood of $\boldsymbol{v}_i$, $\boldsymbol{W}^n \in \mathbb{R}^{d \times d}$. Then each representation $\hat{\boldsymbol{v}}$ is processed by the transformer block networks. Each block applies the following transformations:

$$\tilde{\boldsymbol{v}}_{\boldsymbol{i}} = \text{LayerNorm}(\boldsymbol{v}_i^{'} + \text{LayerNorm}(\hat{\boldsymbol{v}}_{\boldsymbol{i}})), \tag{7}$$

$$\boldsymbol{v}_i^{'} = \text{FFN}(\text{LayerNorm}(\hat{\boldsymbol{v}}_{\boldsymbol{i}})), \tag{8}$$

where FFN(•) is a two layer feedforward network with a non-linear transformation. The transformer block networks are stacked L times to make information propagate through the graph. Finally, we get the graph vertex encodings $\boldsymbol{V}^L = [\boldsymbol{v}_i^L]$ representing the contextualized vertexes and relations in the graph structure.

4.2 Steganographic Text Generation

As we mentioned above, the usual solutions to text generation problems are approaches based on statistical language model [3, 38]. Sentence $\boldsymbol{S}$ can be regarded as a sequential signal and the tth word $Word_t$ is the signal at time step t. This kind of approaches take advantage of RNN's powerful ability in feature extraction and expression for sequential signals. The RNN network can learn the statistical language model from a large number of normal texts and Build a dictionary $\boldsymbol{D}$ of candidate words. When generating, the RNN network calculate the probability distribution of the tth word at time step t according to the previous generated words $\{Word_1, Word_2, \ldots, Word_{t-1}\}$, that is a conditional probability distribution $p(Word_t | Word_1, Word_2, \ldots, Word_{t-1})$.

In this article, the tasks of the text generation model are twofold: (a) introducing topic-specific contents from the knowledge graph and (b) hiding secret information in the generated text. Therefore, We also employ a RNN network with LSTM units [10] as the decoder, but Some improvements were made.

For the first task, to introduce topic-specific contents from the graph, the graph vertex encodings $\boldsymbol{V}^L = [\boldsymbol{v}_i^L]$, we obtained in the last subsection are imported to the LSTM network, besides the $t-1$ previous generated words. Therefore, the conditional probability distribution of the tth word can be rewritten as

$$p(Word_t | Word_1, Word_2, \ldots, Word_{t-1}, \boldsymbol{V}^L). \tag{9}$$

To ensure that the generated text contains the topic-specific contents from the graph, we set up a copy mechanism [9] module in the decoder and a N-head attention layer is used again to fuse the graph vector $\boldsymbol{v}_i^L$ and hidden state of the latest iteration. At each decoding time step t, the graph context vectors $\boldsymbol{c}_t$ is computed using the N-headed attention contextualized by decoder hidden state $\boldsymbol{h}_t$, that is

$$a_j^n = \frac{\exp(a^n(\boldsymbol{h}_t, \boldsymbol{v}_j^L))}{\sum_{j \in \mathcal{N}_i} \exp(a^n(\boldsymbol{h}_t, \boldsymbol{v}_j^L))}, \tag{10}$$

$$\boldsymbol{c}_t = \boldsymbol{h}_t + \|_{n=1}^{N} \sum_{j \in \mathcal{V}'} a_j^n \boldsymbol{W}^n \boldsymbol{v}_j^L. \tag{11}$$

The probability p of copying from the input and the final probability of the next word $p(Word_t)$ are computed by

$$p = \sigma(\boldsymbol{W}_{\text{copy}}[\boldsymbol{h}_t \| \boldsymbol{c}_t] + b_{\text{copy}}), \tag{12}$$

$$p(Word_t) = p \times \alpha_{\text{copy}} + (1 - p) \times \alpha_{\text{dic}}. \tag{13}$$

Here, $\alpha_{\text{dic}} = \text{Softmax}([\boldsymbol{h}_t \| \boldsymbol{c}_t])$ is the probability distribution of the dictionary $\boldsymbol{D}$ and $\alpha_{\text{copy}} = a([\boldsymbol{h}_t \| \boldsymbol{c}_t], \boldsymbol{v}_i^L)$ is the probability distribution of the input using $\boldsymbol{h}_t$ and $\boldsymbol{c}_t$.

For the second task, to hide secret information in the generated text, we need to build a mapping from the secret information bitstream to the word space. So, we add a dictionary coding in the generation decoder based on their conditional probability distribution. At each time step t, select m words with the highest probability from the dictionary D according to the calculated probability distribution of tth word $p(Word_t)$ and put them in the Candidate Pool (CP). Sort m words in the CP in descending order. Then, any effective encoding method can be used to encode the words, For example, the fixed-length coding based on a perfect binary tree, Huffman coding [38] and arithmetic coding [39].

When the words in the CP are all encoded, the process of secret information hiding is to select the corresponding word as the output of the current time according to the binary bitstream of the secret information. If all the binary bitstream is embedded before the text generation is completed, then the model will automatically select the word with the highest probability in the CP. And if the secret message is too long, then another topic-specific graph is needed for the follow-up generation. The capacity m of the CP is related to the embedding rate. Generally, in practice, the capacity $m = 2^k$, where k is the embedding rate. As the embedding rate increases, the capacity m of the CP needs to be larger and it is more likely to choose the word with low probability resulting in poor text quality.

Algorithm details of the proposed information hiding process are shown in Algorithm 1. By using this method, we can generate smooth natural language sentences according to the input secret binary bitstream.

4.3 Information Extraction

Information extraction is the is the reverse operation of information hiding. In our proposed method, the implementation of information extraction can use basically the same process as information hiding. Alice and Bob need to have exactly the same knowledge graph dataset and use the same text generation model with the same parameters and the same information hiding coding method.

Specifically, after receiving the generated steganographic text, Bob first extracts topic keywords from the text and finds the corresponding graph from the dataset. The topic extraction tool we used in our actual experiments is introduced in Section 5.1 and to ensure the uniqueness of the graph Bob find in the dataset, the topic of each graph is different in the graph dataset.

Then Bob use the same graph and RNN network to calculate the conditional probability distribution of each word at each time step. And descending the prediction probability of all the words in the dictionary D to construct the same CP contains $m = 2^k$ words, where k is the embedding rate. Finally, use the same coding method to encode the words in the CP. Since the generated sentence is unique, Bob can accurately extract the hidden bitstream according to the received sentence. Algorithm details of the proposed information extraction process are shown in Algorithm 2.

5 EXPERIMENTS AND ANALYSIS

5.1 Dataset and Implementation Details

To evaluate our topic-aware steganography method, a knowledge graph dataset with different topics is required. In this article, we use the AGENDA dataset [14], a dataset of knowledge graphs paired with scientific abstracts. The AGENDA dataset is built by the information extraction (IE) system from the abstracts of 40K scientific papers from the Semantic Scholar Corpus taken from the proceedings of 12 top AI conferences [1].

We manually filtered 500 knowledge graphs and scientific abstract pairs that are on totally different topics. These are used to build the topic-specific graph space by Alice and Bob. Since Bob

Table 1. The Data Statistics of the AGENDA Dataset

	Abstract	Knowledge Graph
Vocabulary	77K	54K
Tokens	5.8M	1.2M
Entities	—	518K
Average Length	141.2	—
Average Number of Vertices	—	12.42
Average Number of Edges	—	4.43

ALGORITHM 1: Information Hiding Algorithm

```
Input:
  secret binary bitstream: $B = \{1, 0, 1, 1, 0, \dots, 0, 1, 0\}$
  size of Candidate Pool: $m$
  graph space $\mathcal{G}$
Output:
  generated steganography text

Data preparing and Model training;
Select a topic-specific graph $g$ from the graph space $\mathcal{G}$;
Convert $g$ into corresponding graph vectors $\boldsymbol{V}^L$ using the graph embedding network;
while not the end of $B$ do
    if not the end of current paragraph then
        Calculate the probability distribution of the next word according to the words generated
          previously and the graph vectors $\boldsymbol{V}^L$;
        Descending the prediction probability of all the words in the dictionary $D$ and select
          the top $m$ sorted words to construct the CP;
        Construct a binary tree using fixed-length coding method according to the probability
          distribution of each word in the CP and encode the tree;
        Read the secret binary stream, and search from the root of the tree according to the
          encoding rules until the corresponding leaf node is found and output its corresponding
          word;
    else
        Random select another topic-specific graph $g'$ in the graph space $\mathcal{G}$, as the start of the
          next paragraph;
if not the end of current paragraph then
    Select the word with the highest probability outside the CP as the output of current time;
    Select the word with the highest probability at each moment as output until the paragraph
      ends;
return Generated steganography text;
```

need to extract the topic from the received steganographic text and find the unique corresponding graph in the dataset. We need to guarantee the uniqueness of the topic of the graph in the dataset. Statistics of the AGENDA dataset are shown in Table 1. The AGENDA dataset is split into 38,720 training, 500 validation, and 500 test datapoints.

The proposed neural network model is implemented based on Pytorch 1.3.0. The model is trained end-to-end to minimize the negative joint log likelihood of the target text vocabulary and the copied entity indices. We optimize the model using Stochastic Gradient Descent with momentum [21]. The Model is trained for 20 epochs with early stopping based on the validation

ALGORITHM 2: Information Extraction Algorithm

```
Input:
  generated steganography text $Text = \{Word_1, Word_2, \ldots, Word_n\}$
  size of Candidate Pool: $m$
  graph space $\mathcal{G}$
Output:
  secret binary bitstream: $B = \{1, 0, 1, 1, 0, \ldots, 0, 1, 0\}$

Extract topic keywords from $Text$ using a topic extraction tool;
Find the corresponding topic-specific graph $g$ from the graph space $\mathcal{G}$ according to the topic keywords;
Convert $g$ into corresponding graph vectors $\boldsymbol{V}^L$ using the graph embedding network;
for each word $Word_t$ in $Text$ do
    Calculate the probability distribution $p(Word_t|Word_1, Word_2, \ldots, Word_{t-1}, \boldsymbol{V}^L)$ according
      to the previously words and the graph vectors $\boldsymbol{V}^L$ using RNN network;
    Descending the prediction probability of all the words in the dictionary $D$ and select the
      top $m$ sorted words to construct the CP;
    Use fixed-length coding method to encode all the words in the CP;
    if $Word_t$ is in the CP then
        Based on the actual accepted word $Word_t$ at each time step, determine the path from
          the root node to the leaf node;
        According to the tree coding rule, ie, the left side of the child node is 0 and the right
          side is 1, extract the corresponding bitstream and append to $B$;
    else
        The information extraction process ends;
return Extracted secret bitstream $B$;
```

loss. A single-layer LSTMs is used, and we use dropout in self attention layers set to 0.3. The dimensions of hidden states, attentions layer and embedding are fixed at 500. The number of attention heads is set to 4. We stopped the training after 15 epochs (2 days). And we use fixed-length coding in the process of information hiding.

5.2 Evaluation of Topic Correlation

We first evaluate the topic correlation between the generated steganographic text and the input topic-specific graph. Since every graph in our dataset has a corresponding original abstract. By comparing the similarity of the generated steganographic text and the original abstract, we can judge the degree of correlation. Therefore, we use **BLEU** [20] and **METEOR** [6] as the evaluation metrics, which are the standard metrics used in automatic translation tasks. The proposed steganography model is tested under different embedding rate. The metric of embedding rate is Bits/word **(Bpw)**, which is the ratio of message bits encoded to sentence length [39].

The evaluation results are shown in Table 2. We compare our results with RNN-stega model [38], which is also a linguistic steganography model only based on statistical distribution model and repeat the comparison experiments with different embedding rates. Bpw = 0 means that we do not embed any information in the generation process, the text is only generated from the topic-specific graph. As can be seen from the comparison results:

- Compared with the condition when Bpw = 0, the BLEU and METEOR scores drops When information is embedded. But the decline in scores is acceptable, especially when Bpw = 1, the BLEU score only dropped by 7.7% and METEOR score dropped by 4.4%.

Table 2. The Evaluation Results of Topic Correlation

	Our Method		RNN-stega [38]	
Bpw	BLEU	METEOR	BLEU	METEOR
0	14.23	18.56	2.32	6.56
1	13.13	17.74	1.89	5.60
2	10.95	15.72	1.43	6.06
3	8.51	13.53	3.21	4.21

Table 3. The Evaluation Results of Text Quality

Bpw	0	1	2	3
Perplexity	46.73	52.23	106.50	194.54

- As the embedding rate increases, the BLEU and METEOR scores decrease, which is quite reasonable.
- Compared with RNN-stega, both BLEU score and METEOR score of our method are much higher than the scores of RNN-stega at any embedding rate. This shows that the text generated by our method is more similar to the original text.

According to the results of the experiment, the proposed topic-aware steganography method can generate steganographic text that retain the information in the topic-specific graph. The generated steganographic text has consistent contents and topic.

5.3 Evaluation of Text Quality

Then, we evaluate the text quality by testing whether the steganographic text generated by our proposed model are close enough to the natural language on the statistical language model. We use a standard metric for sentence quality testing **Perplexity** [28], which is defined as the average per-word log-probability on the test texts:

$$\begin{aligned} \text{Perplexity} &= 2^{-\frac{1}{n}\log p(S)} \\ &= 2^{-\frac{1}{n}\log p(Word_1, Word_2, \ldots, Word_n)} \\ &= 2^{-\frac{1}{n}\sum_{j=1}^{n}\log p(Word_i | Word_1, Word_2, \ldots, Word_{j-1})}, \end{aligned} \tag{14}$$

where $S = \{Word_1, Word_2, \ldots, Word_n\}$ is the generated sentence and $p(S)$ indicates the probability distribution.

We generate 500 steganographic paragraphs for each embedding rate. The means of the Perplexity of training dataset (Bpw = 0) and the generated text under different embedding rates are tested and the results are shown in Table 3. Based on the results, it can be seen that as the embedding rate increases, the Perplexity gradually increase, which means the statistical language distribution difference between the generated text and the training samples gradually increase. Therefore, the steganography method do affect the quality of the generated text. However, the average Perplexity of the generated steganographic text is close to the training dataset, especially when Bpw = 1. This shows that our proposed steganography method can guarantee the quality of the generated text to a certain extent.

Table 4. The Evaluation Results of Anti-detection

	Accuracy	Precision	Recall	F1-score
Score	0.675	0.742	0.654	0.681

Table 5. Generation Example of The Proposed Method

Knowledge Graph	(semi-supervised learning algorithms USED-FOR classifiers);
	(domain knowledge CONJUNCTION l semi-supervised learning algorithms);
	(semi-supervised learning algorithms USED-FOR labeled data);
	(semi-supervised learning algorithms USED-FOR machine learning);
	(semi-supervised learning algorithms USED-FOR natural language processing)

Secret Information Bitstream	[1000111001110101100101011100000001111011011011001010 ...]
1 Bpw	this paper presents a semi-supervised learning algorithms to the problem of learning a classifiers for structured learning tasks. we have designed and evaluated our proposed semi-supervised learning algorithms in this domain ...
2 Bpw	the performance and quality semi-supervised learning algorithms can benefit machine learning. this paper proposes semi-supervised learning algorithms to address this challenge, by introducing new semi-supervised learning algorithms ...
3 Bpw	we investigate whether there in training data are from training images and the data collection, can learn classifiers, or by the knowledge transfer itself can benefit when to be applied to other tasks such to improve natural language processing across a domain that ...

5.4 Evaluation of Anti-Detection

Furthermore, we tested the anti-detection ability of our proposed method. A text steganalysis algorithm [33] is used to classify the generated steganographic sentences. The results of the classification when Bpw = 2 are listed in Table 4. The test results show that our proposed steganography method has a certain ability to resist detection.

5.5 Generation Example

Table 5 displays some examples of the generated steganographic text by the proposed steganography system for a particular test instance. Given the topic-specific graph whose topic keyword is “semi-supervised learning algorithm” in this test instance and the secret information that needs to be hided, three steganographic texts are generated at different embedding rate. We can see that the proposed method makes use of entities from the input graph to ensure the topic.

6 CONCLUSION

We proposed a topic-aware neural linguistic steganography method that can generate a steganographic paragraph with a specific topic. By encoding entities and relationships data from Knowledge Graphs, the text generated by the model has the same associated semantics and topic as the knowledge graph. The contributions of this article are twofold. On the one hand, we use graph-structured data as the source of steganographic text for the first time and achieve the transformation from graph to text. On the other hand, we implement a topic-controllable steganographic long text generation method to guarantee consistent context and better concealment. The experimental results illustrate that the steganographic text generated by the proposed method has consistent contents and topic. And the good performance of the model in text quality and anti-detection ability proved that it is effective and practical. In fact, because the graph-based structure is a more convenient form to describe knowledge, events and relations, the proposed method can be more widely used in covert communication, privacy protection and many other areas of information security. For the future work, We expect to extend the proposed method to larger knowledge graphs.

REFERENCES

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Michael Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler C. Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna L. Power, Sam Skjonsberg, Lucy Lu Wang, Christopher Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'18)*.

[2] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2000. A neural probabilistic language model. *J. Mach. Learn. Res.* 3 (2000), 1137–1155.

[4] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *R*etrieved from https://arxiv.org/abs/1904.10509.

[5] Nopporn Chotikakamthorn. 1998. Electronic document data hiding technique using inter-character space. In *Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems. Microelectronics and Integrating Systems.* 419–422.

[6] Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Workshop on Statistical Machine Translation (WMT@ACL'14)*.

[7] Abdelrahman Desoky. 2010. Comprehensive linguistic steganography survey. *Int. J. Info. Comput. Secur.* 4, 2 (2010), 164–197.

[8] Jessica Fridricha. 2009. *Steganography in Digital Media: Principles, Algorithms, and Applications.* Cambridge University Press, Cambridge, UK.

[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. Retrieved from https://arxiv.org/abs/1603.06393.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9 (1997), 1735–1780.

[11] Christophe Guyeux, Jean F. Couchot, and Raphael Couturier. 2015. STABYLO: Steganography with adaptive, Bbs, and binary embedding at low cost. *Ann. Telecommun.* 70, 9–10 (2015), 441–449.

[12] Lucai Wang Jianjun Zhang, Jun Shen, and Haijun Lin. 2016. Coverless text information hiding method based on the word rank map. In *Proceedings of the International Conference on Cloud Computing and Security*, Vol. 10039. Springer, Cham.

[13] Lucai Wang Jianjun Zhang, Yicheng Xie, and Haijun Lin. 2017. Coverless text information hiding method using the frequent words distance. In *Proceedings of the International Conference on Cloud Computing and Security*, Vol. 10602. Springer, Cham.

[14] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers. Retrieved from https://arxiv.org/abs/1904.02342.

[15] Chunfang Yang, Lingyun Xiang, Xinhui Wang, and Peng Liu. 2017. A novel linguistic steganography based on synonym run-length encoding. *IEICE Trans. Info. Syst.* 100, 2 (2017), 313–322.

[16] Gang Luo, Lingyun Xiang, Xingming Sun, and Bin Xia. 2014. Linguistic steganalysis using the features derived from synonym frequency. *Multimedia Tools Appl.* 71, 3 (2014), 1893–1911.

[17] Anandaprova Majumder and Suvamoy Changder. 2013. A novel approach for text steganography: Generating text summary using reflection symmetry. *Procedia Technol.* 10, 10 (2013), 112–120.

[18] H. Hernan Moraldo. 2014. An approach for text steganography based on markov chains. Retrieved from https://arxiv.org/abs/1409.0915.

[19] Brian Murphy and Carl Vogel. 2007. The syntax of concealment: Reliable methods for plain text information hiding. In *Proceedings of the SPIE*, Vol. 6505. Springer, Cham, 752–762.

[20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics (ACL'01).*

[21] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural Netw.: Offic. J. Int. Neural Netw. Soc.* 12 1 (1999), 145–151.

[22] Alec Radford. 2018. Improving language understanding by generative pre-training.

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

[24] Claude E. Shannon. 1949. Communication theory of secrecy systems. *Bell Syst. Tech. J.* 28 (1949), 656–715.

[25] A. N. Shniperov and K. A. Nikitina. 2016. A text steganography method based on Markov chains. *Autom. Control Comput. Sci.* 50 (2016), 802–808.

[26] Gustavus J. Simmons. 1983. The prisoners' problem and the subliminal channel. In *Proceedings of the International Cryptology Conference (CRYPTO'83).*

[27] Dilip K. Yadav Susmita Mahato, and Danish A. Khan. 2020. A modified approach to data hiding in microsoft word documents by change-tracking technique. *J. King Saud Univ. Comput. Info. Sci.* 32 (Feb. 2020), 216–224.

[28] Martin Jaggi, Tina Fang, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. *Commun. ACM* (May 2017). Retrieved from https://arxiv.org/abs/1705.10742.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS'17).*

[30] Yonghui Dai, Weihui Dai, Yue Yu, and Bin Deng. 2010. Text steganography system using Markov chain source model and des algorithm. *J. Softw.* 5, 7 (2010), 785–792.

[31] Andreas Westfeld and Andreas Pfitzmann. 1999. Attacks on steganographic systems. In *Information Hiding.*

[32] Zhongliang Yang, Yuting Hu, Yongfeng Huang, and Yujin Zhang. 2019. Behavioral security in covert communication systems. Retrieved from https://arxiv.org/abs/1910.09759.

[33] Zhongliang Yang, Yongfeng Huang, and Yu-Jin Zhang. 2019. A fast and efficient text steganalysis method. *IEEE Signal Process. Lett.* 26 (2019), 627–631.

[34] Jian Yuan, Yongfeng Huang, and Shanyu Tang. 2011. Steganography in inactive frames of VoIP streams encoded by source codec. *IEEE Trans. Info. Forensics Secur.* 6, 2 (June 2011), 296–306.

[35] Fufang Li, Yubo Luo, Yongfeng Huang, and Chinchen Chang. 2016. Text steganography based on ci-poetry generation using Markov chain model. *KSII Trans. Internet Info. Syst.* 10, 9 (2016), 4568–4584.

[36] Rohan Harit, Xianyi Chen, Zhili Zhou, Huiyu Sun, and Xingming Sun. 2015. Coverless image steganography without embedding. In *Proceedings of the International Conference on Cloud Computing and Security.* Springer, Cham, 123–132.

[37] Yongfeng Huang, Zhongliang Yang, and Xueshun Peng. 2017. A sudoku matrix-based method of pitch period steganography in low-rate speech coding. In *Proceedings of the International Conference on Security and Privacy in Communication Systems.* Springer, Cham, 752–762.

[38] Ziming Chen, Yongfeng Huang, Zhongliang Yang, Xiaoqing Guo, and Yu-Jin Zhang. 2018. RNN-Stega: Linguistic steganography based on recurrent neural networks. *IEEE Trans. Info. Forensics Secur.* (Sep. 2018), 1280–1295. DOI: https://doi.org/10.1109/TIFS.2018.2871746

[39] Zachary M. Ziegler, Yuntian Deng, and Alexander M. Rush. 2019. Neural linguistic steganography. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP/IJCNLP'19).*

Received February 2020; revised June 2020; accepted August 2020