Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065804

ARTICLE

Linguistic Steganography Based on Sentence Attribute Encoding

Lingyun Xiang*, Xu He, Xi Zhang and Chengfu Ou

School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, 410114, China

*Corresponding Author: Lingyun Xiang. Email: xiangly210@163.com

Received: 21 March 2025; Accepted: 13 May 2025; Published: 03 July 2025

ABSTRACT: Linguistic steganography (LS) aims to embed secret information into normal natural text for covert communication. It includes modification-based (MLS) and generation-based (GLS) methods. MLS often relies on limited manual rules, resulting in low embedding capacity, while GLS achieves higher embedding capacity through automatic text generation but typically ignores extraction efficiency. To address this, we propose a sentence attribute encoding-based MLS method that enhances extraction efficiency while maintaining strong performance. The proposed method designs a lightweight semantic attribute analyzer to encode sentence attributes for embedding secret information. When the attribute values of the cover sentence differ from the secret information to be embedded, a semantic attribute adjuster based on paraphrasing is used to automatically generate paraphrase sentences of the target attribute, thereby improving the problem of insufficient manual rules. During the extraction, secret information can be extracted solely by employing the semantic attribute analyzer, thereby eliminating the dependence on the paraphrasing generation model. Experimental results show that this method achieves an extraction speed of 1141.54 bits/sec, compared with the existing methods, it has remarkable advantages regarding extraction speed. Meanwhile, the stego text generated by this method respectively reaches 68.53, 39.88, and 80.77 on BLEU, △PPL, and BERTScore. Compared with the existing methods, the text quality is effectively improved.

KEYWORDS: Linguistic steganography; paraphrase generation; semantic attribute

1 Introduction

Steganography is the art and science of hiding information in ordinary media without arousing suspicion from supervisors [1]. Unlike encryption techniques [2,3] that convert a piece of plaintext into ciphertext, steganography can conceal secret information in any public multimedia carrier with redundant space, such as texts [4,5], images [6–8], audio [9,10], and video [11,12]. Among them, texts are the most frequently communicated medium due to their high universality and robustness [13]. Therefore, linguistic steganography [14] employing texts as the carriers has received widespread attention and research.

Existing linguistic steganography mainly focuses on two categories: modification-based linguistic steganography (MLS) and generation-based linguistic steganography (GLS). MLS primarily processes semantic equivalence transformations at the lexical and sentence levels to implement secret information hiding. From a lexical perspective, relying on the synonyms library, suitable synonyms can be selected to perform substitution operations to achieve information hiding [15–17]. However, such methods often need to construct appropriate synonyms and complex optimization methods to ensure that the statistical characteristics of the replaced sentences are undisturbed. At the sentence level, syntactic rules such as syntactic transformations [18–20] and paraphrasing [21,22], etc., can be used to construct specific templates

to embed secret information. Such methods can maintain semantic invariance to the greatest extent and have strong anti-steganalysis capabilities. However, these methods rely heavily on artificially designed rules, which may only apply to specific text types. For different types of text, new rules need to be redesigned, which will consume a lot of human resources. Moreover, artificially designed rules are limited by the designer's language level and the development level of natural language processing technology at that time, resulting in the problem of fewer rules.

In recent years, GLS has begun to prevail [23–26], which is based on neural network language models (LMs) to automatically generate stego text that conforms to statistical characteristics. They have made great progress in embedding capacity, providing a new idea for improving the performance of linguistic steganography. However, as the length of secret information increases, they may select inappropriate candidate words, resulting in the contextual semantic inconsistency of stego text [27]. To this end, some recent studies try to introduce semantically relevant constraints [28–30], thereby improving the semantic consistency of stego text. Unfortunately, the above GLS methods only consider the sender's ability to hide information, ignoring the receiver's resource limitation and the complexity of extracting secret information. While achieving high performance, they often need to share more resources required for embedding, which brings massive resource consumption to the receiver to extract secret information.

To overcome the limitations mentioned above, we introduce the LMs into MLS and propose a novel linguistic steganography based on sentence attribute encoding. This method uses the paraphrase generation model to automatically generate various semantically similar sentences (paraphrase sentences) by learning the features of cover sentences, thereby reducing the dependence on manual design rules. Considering that traditional steganographic coding will fail to extract secret information due to the difference between the cover sentences and paraphrase sentences, we start from the intrinsic characteristics of the sentence, according to the attributes of these two types of sentences for steganographic coding. Specifically, we construct a lightweight semantic attribute analyzer to learn the deep semantic features of sentences, then classify them according to the differences between the two types of sentences, and finally assign corresponding semantic attribute values to the sentences and encode them. During the extraction, the receiver can extract secret information solely by relying on the semantic attribute analyzer to acquire sentence attributes, without sharing the identical generation model and other resources with the sender. Consequently, this method effectively eliminates the dependence on the generation model.

In summary, our contributions are as follows:

(1) We proposed a novel linguistic steganography based on sentence attribute encoding that expands the number of equivalent substitution sentences by training a semantic attribute adjuster, overcoming the reliance on cumbersome manual features.

(2) We design a simple semantic attribute analyzer as a proxy model to address the reliance on language models in the information extraction process.

(3) Experimental results show that the proposed method has better robustness and higher extraction efficiency compared to baselines, and the stego text has better text quality.

2 Related Work

Modification-based linguistic steganography (MLS) relies on various semantically equivalent language conversion methods to hide secret information while ensuring that the global or local semantics of the text remain unchanged. Currently, relatively mature work mainly includes methods based on synonym substitution, syntactic transformation, and paraphrasing.

2.1 Synonym Substitution

The synonym substitution-based MLS usually selects similar words to replace the original words to hide secret information. To select appropriate synonyms, Bolshakov and Gelbukh [31] proposed using the synonym dictionary of WordNet and combining it with statistical information collected from the Internet to determine whether an effective collocation is formed for synonym substitution. Chang and Clark [15] advocated using the Google n-gram corpus to check the contextual applicability of synonyms. In addition, a vertex color coding method is created to give each synonym a unique coding value to resolve the ambiguity caused by words with multiple meanings when encoding. However, these methods may destroy the statistical characteristics of the text. Therefore, Xiang et al. [32] studied a synonym run-length encoding method, advocating to balance the distribution of synonyms with different frequencies through adaptive positive and negative synonym conversion. Li et al. [17] considered using the conditional probability of the source word to control the joint probability so that the frequency distribution of synonyms in the stego text is the same as that in the normal corpus. Moreover, to improve the anti-steganalysis ability and embedding capacity, Xiang et al. [33] combined compression and selection strategies to reduce actual embedding operations to improve the embedding ability and selected the best text with high imperceptibility according to a given rule derived from the distance between natural text and stego text. Mahato et al. [34] used non-occurrence probability instead of occurrence probability to construct the Huffman tree, thereby improving the embedding capacity.

2.2 Syntactic Transformation

The syntax transformation-based MLS primarily achieves information hiding by altering syntactic structures. Murphy and Vogel [20] proposed a set of automated reversible syntactic transformations that can hide information without changing the meaning or style of the text. Chang and Clark [35] generated multiple candidate texts using word ranking techniques. He et al. [36] proposed a hybrid steganography system that uses a syntactic analyzer to determine the syntactic structure of the text. Kim and Goebel [37] performed syntactic dependency analysis on cover text based on the syntactic dependency tree and segmented syntactic tree to determine the syntactic structure of the text.

2.3 Paraphrasing

The paraphrasing-based MLS mainly uses annotations in the dictionary, paraphrase template rules, etc., to rewrite the text content to embed secret information. Stutsman et al. [38] used multiple translation systems to obtain enough sentences and then selected different translation sentences to hide secret information. Meng et al. [39] selected one from the n-best list for each cover sentence to hide the information. Chang and Clark [21] proposed to hide information by using a large paraphrase dictionary containing many intensive paraphrase rules and used the Google n-gram corpus and CCG parser to verify paraphrasing grammaticality and fluency. Wilson and Ker [22] adopt paraphrase rules and use distortion measures to automatically generate the best-embedded stego text. Yang et al. [40] propose a pivot translation-based paraphrasing method, which designs a semantic-aware bins coding strategy to transform the expression of given cover text, thereby generating stego text.

3 Proposed Method

3.1 Overall Architecture

The proposed linguistic steganography based on sentence attribute encoding is a sentence-level MLS. Its core is analyzing the sentence attribute and automatically generating alternative candidate sentences to embed secret information by using the paraphrase generation model.

As illustrated in Fig. 1, our method takes the sentence as a unit, and the embedding process of each sentence in the text is the same. Let sentence $S_a = \{x_1, x_2, \cdots, x_n\}$ be any original sentence in cover text T. To start with, S_a is sent to the semantic attribute analyzer $\mathcal{D}$ to get its semantic attribute value $B(S_a)$, and compare with the current secret information Q_a to be embedded. If $B(S_a) = Q_a$, then S_a is the sentence that embeds secret information Q_a. If $B(S_a) \neq Q_a$, then input S_a into the sentence semantic attribute adjuster $\mathcal{G}$, generating the paraphrase sentence S_a' with semantic attribute value equal Q_a as the stego sentence.

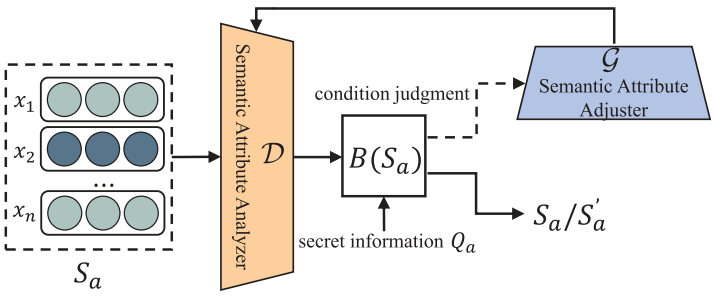


Figure 1: Information embedding process of the proposed method

For instance, assuming the current bit embedded is '1', the cover sentence is "I like strawberries best.", and its semantic attribute value is '0', this sentence cannot embed the bit '1'. At this time, we input the cover sentence into $\mathcal{G}$ to generate the paraphrase sentence "My favorite fruit is strawberry." with a semantic attribute value of '1' as a stego sentence, thereby embedding the bit '1'.

3.2 Semantic Attribute Analyzer

The sentence semantic attribute analyzer analyzes the intrinsic characteristics of sentences, assigning different attribute values for cover and semantically similar paraphrasing sentences, thereby establishing a reversible mapping relationship between sentences with different attributes and secret information. To better distinguish between the cover and paraphrase sentences, we use a deep neural network to build a semantic analyzer, which is mainly divided into two parts: dense matrix acquisition and semantic feature extraction and classification.

Firstly, the embedding matrix of the sentence is converted into a dense matrix representation through a feedforward neural network to be used as model input; Secondly, the semantic feature vector of the sentence can be obtained by learning the semantic features using different convolution kernels, and then the semantic attribute value of the sentence can be obtained by mapping the semantic feature vector into a two-dimensional probability space through projection and normalization functions.

3.2.1 Dense Matrix Acquisition

Let any cover sentence represent $S_a = \{x_1, x_2, \ldots, x_n\}$ in cover text $T = \{S_1, S_2, \cdots, S_m\}$, where x_i represents the i-th word in S_a and x_n denotes the n-th word in S_a. We use one-hot encoding to represent the sentence S_a as an embedding matrix containing only 0 and 1. First of all, we count the word frequency of the words in the corpus and then arrange the words in descending order according to the word frequency, thereby constructing a vocabulary $V = [v_1, v_2, \cdots, v_M]$, where v_j denotes the j-th word in V and M indicates the vocabulary size. Then, the word x_i is encoded by the one-hot encoding, resulting in a vector expression of the word as $c_i = \left[c_i^1, c_i^2, \cdots, c_i^M\right]$, where $c_i \in \{0,1\}^{1\times M}$, and $c_i^j = 1$ only when $x_i = v_j$, otherwise the other

elements in c_i are equal to zero, i.e., $c_i^j = 0$. Ultimately, after encoding all the words in the sentence S_a, the embedding matrix C of S_a can be obtained.

It is worth noting that matrix C is a sparse matrix with a large amount of redundant information, and the information of each word is relatively isolated, making it difficult to reflect the contextual dependencies between words. Therefore, we transform the sparse matrix into a dense matrix to integrate the semantic information in the sentence, which lays the foundation for the semantic feature extraction and attribute classification in subsequent sentences.

We use a layer of feedforward neural network f to transform the matrix C into a dense matrix $A \in R^{n \times d}$, calculated as follows:

$$A^T = f(c) = W_1^T \cdot C + b_1 \tag{1}$$

where A denotes dense matrix of sentence S_a, A^T indicates transpose of A, W_1 represents weight matrix of f, which is initialized by random sampling from uniform distribution $[-1, 1]$, W_1^T denotes transpose of W_1, $W_1 \in R^{M \times d}$, R represents the set of real numbers, d denotes dimension of W_1, $b_1 \in R$ indicates bias of f.

3.2.2 Semantic Feature Extraction and Classification

To extract features from the dense matrix A, we use a one-dimensional convolutional neural network to capture deep semantic features in sentences. Specifically, we set the weight matrix of the convolution kernel with kernel size t to act on A, and the number of convolutional kernels is set to K. The convolutional kernel will perform convolution operation by moving window size t along the length of the sentence, thereby acquiring the potential feature vector $H_t \in R^{(n-t+1) \times d}$ though capture context dependence in the sentence. The potential eigenvectors are calculated as follows:

$$\begin{aligned} H_t &= [h_1, h_2, \cdots, h_{n-t+1}] \\ h_i &= \left[h_i^1, h_i^2, \cdots, h_i^K\right], i = 1, 2, \cdots, n-t+1 \\ h_i^k &= \mathrm{Re}\,lu(w_t \otimes A[i : i+t-1] + b_t), k = 1, 2, \ldots, K \end{aligned} \tag{2}$$

where h_i represents eigenvector of i-th position, h_i^k denotes output of i-th position and k-th convolution kernel, $\otimes$ indicates convolution operation, $b_t \in R$ denotes bias of convolution kernel.

Considering the context dependency distances differ between words, we set up convolutional kernels with three kernel sizes, $t = 3$, $t = 5$, and $t = 7$, to extract multi-granular semantic features from different window distances. Subsequently, these features are averagely pooled and spliced to adapt to different sentence lengths. The calculation process is as follows:

$$E_t = MeanPool(H_t) = \frac{1}{n-t+1} \sum_{i=1}^{n-t+1} h_i, t = 3, 5, 7 \tag{3}$$

$$E = E_3 \oplus E_5 \oplus E_7 \tag{4}$$

where $E_t \in R^K$ represents the semantic feature vector of S_a obtained after average pooling, $\oplus$ denotes concatenation operation, $E \in R^{3K}$ indicates semantic feature vector of sentence obtained after concatenation.

After obtaining the final semantic feature vector of the sentence E, to classify the sentence attributes, we first use a projection function $\mathcal{F}(\cdot)$ to project E into a two-dimensional space. The calculation is as follows:

$$L = \mathcal{F}(E) = W_2^T \cdot E + b_2 \tag{5}$$

where $W_2 \in R^{3K\times 2}$ and $b_2 \in R$ respectively denotes the weights and biases of the $\mathcal{F}(\cdot)$. The W_2 is initialized using the He initialization [41], $L \in R^2$ is a logic vector in two-dimensional space.

Subsequently, the values of each dimension of L are mapped to the probability space $[0,1]$ using the $softmax(\cdot)$ function, and the calculation is as follows:

$$p(\xi|S_a) = softmax(L) = \frac{e^{L_\xi}}{e^{L_1} + e^{L_2}}, \xi = 0,1 \tag{6}$$

Ultimately, according to the classification results, the semantic attribute value of the sentence can be obtained by using the $argmax(\cdot)$ function to take the dimension corresponding to the maximum probability value. The calculation is as follows:

$$B(S_a) = \underset{\xi}{argmax}(p(\xi|S_a)) \tag{7}$$

When the dimension corresponding to the maximum probability value is the first dimension, the semantic attribute value of the sentence $B(S_a) = 0$; otherwise, $B(S_a) = 1$.

3.3 Semantic Attribute Adjuster Based on Paraphrase Generation

If the secret bit $Q_a \in \{0,1\}$ to be embedded in the current sentence is inconsistent with $B(S_a)$, it means that S_a cannot embed Q_a. In this case, we will use the semantic attribute adjuster to generate a new paraphrase sentence S'_a with a semantic attribute value equal to Q_a to replace cover sentence, thereby achieving adjust for the sentence attribute value.

The semantic attribute adjuster uses the conditional variational autoencoder (CVAE) as the basic framework and adopts the corpus of cover and paraphrase sentences for training. To start with, we use Word2Vec to acquire a word vector representation of each word in the sentence:

$$o_i = Word2Vec(x_i) \tag{8}$$

where $o_i \in R^{d_o}$ represents the word vector to the i-th word x_i, d_o denotes the dimension of o_i.

Such a word vector contains rich contextual knowledge. The vector of the corresponding sentence is expressed as $O = [o_1, o_2, \cdots, o_n]$, $O \in R^{n\times d}$, where O indicates the sentence vector after n-word vectors are spliced, n represents the number of words in the sentence S_a.

Subsequently, the GRU neural network is used as the sentence encoder of CVAE, and the distance dependence between sentence sequences is captured by using the reset gate r_t and the update gate u_t. The calculation procedures are as follows:

$$r_t = sigmoid([h_{t-1}, o_t]W_r) \tag{9}$$

$$u_t = sigmoid([h_{t-1}, o_t]W_u) \tag{10}$$

$$\hat{h_t} = Tanh([r_t \odot h_{t-1}, o_t]W_h) \tag{11}$$

$$h_t = h_{t-1}(1 - u_t) + \hat{h_t} \odot u_t \tag{12}$$

where $sigmoid(\cdot)$ and $Tanh(\cdot)$ denote the activation function, o_t is the word vector representation of t-th word in S_a, h_{t-1} denotes the hidden state of the previous word of the t-th word, W_r and W_u represents the weight of reset and update gate, W_h indicates the weight matrix for fuse the information of h_{t-1}, h_t represents

the hidden state after the reset, $\odot$ denotes the dot-product operation, h_t denotes the updated hidden state. When $t = 1$, the initial state h_0 is initialized to a zero vector. The above operation is expressed as follows:

$$h_t = GRU(h_{t-1}, o_t) \tag{13}$$

The word vector o_t of the t-th word in the vector O thus obtains the hidden state information h_t of the t-th word after passing through the gated recurrent unit neural networks. When $t = n$, the hidden state vector h_n of the sentence S_a can be obtained. This hidden state vector contains the context dependency of all words in S_a, ensuring that a natural and fluent sentence can be generated in a subsequent process.

Ultimately, the GRU neural network is used as the decoder of CVAE, and the hidden state vector h_n of sentence S_a is used as the control condition to control the decoder to generate new sentences. The specific calculation process is as follows:

$$l_t = GRU(h_n, o_{t-1}^y \oplus z) \tag{14}$$

$$y_t = argmax(Softmax(W_y^T l_t)) \tag{15}$$

$$o_{t-1}^y = Word2Vec(y_{t-1}) \tag{16}$$

$$z \sim N(0,1) \tag{17}$$

where l_t represents the hidden state of the current generated word, the initial hidden state is h_n, o_{t-1}^y denotes the word vector of the previous generated word, y_t indicates the subscript of the t-th generated word in the vocabulary V, $GRU(\cdot)$ is gated recurrent neural network, W_y represents the projection matrix, z is vector sampled from standard normal distribution.

The sampled z can reconstruct the sentence in the corpus, using h_n as a control condition so that z can generate a new sentence S_a, which is semantically similar or identical. If it is not equal to Q_a, it will continue to generate a new sentence $S'_a = y_1, y_2, \cdots, y_N$ so that $B(S'_a) = Q_a$, where y represents the generated word and N denotes the length of S'_a.

3.4 Algorithm

We construct a semantic analyzer and a semantic adjuster to realize the embedding and extraction of secret information. The specific embedding and extraction algorithms (Algorithms 1 and 2) are as follows.

Algorithm 1: Embedding algorithm

Input:

Cover text $T = \{S_1, S_2, \ldots, S_m\}$, Secret information to be embedded $Q = \{Q_1, Q_2, \ldots, Q_m\}$, Semantic attribute analyzer $\mathcal{D}$, Semantic attribute adjuster $\mathcal{G}$.

Output:

Stego text $T' = \{S'_1, S'_2, \ldots, S'_m\}$

1. Initialize $T' = \{\}$
2. **for** $a = 1$ to m **do**
3. Define the current secret bit to be embedded as Q_a and the current cover sentence as S_a
4. Obtain the attribute value $B(S_a) = \mathcal{D}(S_a)$ of S_a using semantic attribute analyzer $\mathcal{D}$
5. **if** $B(S_a) = Q_a$ **then**

(Continued)

Algorithm 1 (continued)

It means that the current cover sentence S_a can embed Q_a, $T' = T' + \{S_a\}$
end if
while $B(S_a) \neq Q_a$ **do**
Generate a paraphrase sentence S'_a through input S_a to semantic attribute adjuster $\mathcal{G}$
Obtain the attribute value $B(S'_a) = \mathcal{D}(S_a)$ of S'_a using semantic attribute analyzer $\mathcal{D}$
if $B(S'_a) = Q_a$ **then**
$T' = T' + \{S'_a\}$
end if
end while
end for
return the stego text T'

Algorithm 2: Extraction algorithm

Input:

Stego text $T' = \{S'_1, S'_2, \ldots, S'_m\}$, Semantic attribute analyzer $\mathcal{D}$.

Output:

Secret information $Q = \{Q_1, Q_2, \ldots, Q_m\}$.

Initialize $Q = \{\}$
for $a = 1$ to m **do**
Define the current stego sentence as S_a
Obtain the attribute value $Q_a = \mathcal{D}(S'_a)$ of S'_a using semantic attribute analyzer $\mathcal{D}$
$Q = Q + \{Q_a\}$
end for
return the secret information $Q = \{Q_1, Q_2, \ldots, Q_m\}$

4 Experimental Results and Analysis

4.1 Datasets

In the experiments, we utilized two widely used datasets, Quora[1] and MSCOCO [42]. The Quora dataset contains approximately 400,000 question pairs labeled with a binary value, where 1 indicates semantic equivalence despite differing expressions, and 0 denotes different semantics. We selected only the question pairs labeled as 1 for our experiments. MSCOCO is a large-scale image recognition dataset with over 120,000 human-annotated captions, each associated with five captions written by different annotators, capturing similar semantic content. For training, 111,715 Quora question pairs and 200,815 MSCOCO captions were selected, with 3000 samples for validation and 20,000 for testing in each dataset.

Prior to model training, the datasets were preprocessed by truncating sentences longer than 30 words, converting uppercase letters to lowercase, and maintaining a vocabulary of 25k. The preprocessed data was first used to train the paraphrase generation model, which then generated paraphrase sentences for the training and validation sets. Subsequently, cover and paraphrase sentences were employed to train the sentence semantic attribute analyzer, with cover sentences labeled as '1' and paraphrase sentences as '0'. The test set was used to evaluate method performance.

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs (accessed on 12 May 2025)

4.2 Model Training Settings

(1) Semantic Attribute Adjuster Model Training Settings

We use the open-source pre-trained model Word2Vec [43] to obtain the word vector with dimension $d_0 = 300$. Both the encoder and decoder use 3-layer stacked GRU neural networks, set the number of hidden layer units in each layer to 512, and set the dimension of potential encoding z to 128. The batch size of the model is 50, and the total number of training epochs is 30. In the model training stage, the Adam optimizer calculates the model gradient and updates the parameters, and the initial learning rate is set at 0.001.

(2) Semantic Attribute Analyzer Model Training Settings

We set the output vector dimension of the feedforward neural network f to $d = 300$ and the number of convolutional kernels $K = 128$. The batch size is 128, and the total number of training epochs is 30. In the model training stage, the Adam optimizer calculates the model gradient and updates the parameters, and the initial learning rate is set at 0.001.

4.3 Baselines and Metrics

We compare a typical generation-based linguistic steganography RNN-stega [23] and an advanced linguistic steganography based on paraphrasing SPLS [40]. For the performance evaluation of methods, we test from four aspects: extraction efficiency, text quality, anti-steganalysis, and robustness. To evaluate the extraction efficiency, we define a metric called BT, which is obtained by dividing the total number of bits by the total time. The larger, the better. For the text quality, we measure text similarity, text fluency, and semantic consistency of the generated stego texts by employing BLEU [44], △PPL, and BERTScore [45] as metrics. Among them, the △PPL represents the difference in perplexity (PPL) [23] between cover and stego texts. The calculation is as follows:

$$\triangle PPL = |PPL_{stego} - PPL_{cover}| \tag{18}$$

To assess the anti-steganalysis ability, we select TS-RNN [46] and LS-CNN [47], two steganalysis methods to distinguish stego from cover texts. The detection Accuracy (ACC), F1-score (F1), Precision (Pr), and Recall (Re) are employed as metrics for evaluating the anti-steganalysis ability of our method. To test the robustness, we simulate three attack methods: synonym substitution, delete, and insert, and calculate the extraction success rate of secret information after each attack method. The extraction success rate is defined as $\frac{K}{N}$, where K is the number of sentences or words successfully extracting secret information, N is the number of all sentences or words embedding secret information.

4.4 Results and Analysis

(1) Extraction Efficiency Analysis

Under the same embedding capacity condition, we set the number of candidate words selected as 2, 4, and 8 for VLC in RNN-stega, corresponding to VLC-1, VLC-2, and VLC-3. For the FLC in RNN-stega, the embedding rates are set to 1, 2, and 3, corresponding to FLC-1, FLC-2, and FLC-3, respectively. The comparative results of the extraction efficiency experiments are shown in Table 1.

Table 1: Experimental results of extraction efficiency. The best is highlighted in bold

Methods	BT (bits/s) ↑	Model size ↓
RNN-stega (FLC-1)	506.36	53.7 M
RNN-stega (FLC-2)	972.84	
RNN-stega (FLC-3)	**1379.99**	
RNN-stega (VLC-1)	502.27	
RNN-stega (VLC-2)	801.19	
RNN-stega (VLC-3)	1122.26	
SPLS	745.73	63.4 M
Ours	**1141.54**	**33.3 M**

As shown in Table 1, the model size used for RNN-stega and SPLS is 53.7 M and 63.4 M, and our semantic attribute analyzer is only 33.3 M. Although our method embeds secret information in sentence units, its extraction efficiency reaches 1141.54 bits/second, which is much higher than the extraction efficiency of VLC-1, VLC-2, FLC-1, FLC-2, and SPLS. This is because our method does not need to fully share all the resources required for embedding during the extraction process. Moreover, the extraction efficiency of our method is comparable to that of VLC-3 and FLC-3. This is because our method embeds 1 bit per sentence, while the FLC-3 embeds 3 bits per word. In the case of embedding the same secret information, the more secret information embedded in a word, the fewer times the model runs during extraction. For generation-based linguistic steganography, the most direct way to improve performance is to increase the number of parameters of the generative model or use a larger model. However, for the receiver, the resource consumption will also increase, and the sender and receiver will also re-share the key; otherwise, the secret information cannot be extracted. Our method allows the sender to update the generative model while keeping the analyzer unchanged, without affecting the receiver to extract the secret information.

(2) Text Quality Analysis

As show in Table 2, for both BLEU and BERTScore, our method far exceeds FLC-3, VLC-3, and SPLS on both datasets. This is because the stego text of the FLC-3 and VLC-3 is generated under the guidance of secret information. The semantics of stego text differ greatly from those of naturally generated text without embedded secret information. For SPLS, this method changes the expression of the text by translation, which may cause some semantic differences before and after translation. Our method retells the cover sentence only when the attribute value of the cover sentence is inconsistent with the bit to be embedded, thereby reducing the impact on the sentence semantics. In addition, the △PPL of our method is much lower than that of VLC-3, FLC-3, and SPLS, which indicates that the stego text generated by our method can better maintain the fluency of the cover text. In contrast, the text semantics of the generative method have a long-distance dependence, and the text context relevance weakens with the increase in text length, resulting in lower fluency.

Table 2: Experimental results of text quality. The best is highlighted in bold

Dataset	Methods	BLEU ↑	△PPL ↓	BERTScore ↑
Quora	RNN-stega (FLC-3)	8.77	307.66	27.05
	RNN-stega (VLC-3)	9.01	146.76	26.60
	SPLS	25.79	81.61	63.14
	Ours	**68.53**	**39.88**	**80.77**

(Continued)

Table 2 (continued)

Dataset	Methods	BLEU ↑	△PPL ↓	BERTScore ↑
MSCOCO	RNN-stega (FLC-3)	10.52	213.56	39.12
	RNN-stega (VLC-3)	10.19	82.15	39.23
	SPLS	17.63	70.67	65.23
	Ours	**59.54**	**62.23**	**75.68**

(3) Anti-Steganalysis Ability Analysis

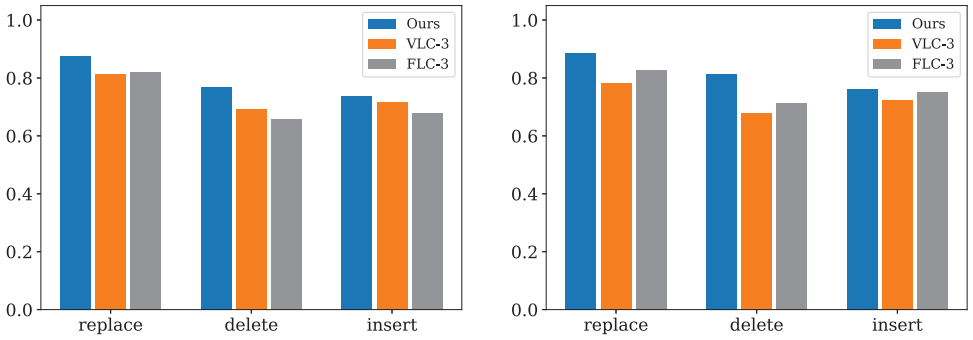
Table 3 shows that in two datasets and bits per word (bpw), the accuracy of the two steganalysis methods is below 70%, while the recall rate is basically about 50%. This shows that the secret information from the stego text generated by our method can hardly be detected. Therefore, our method can ensure the safe transmission of secret information and has high anti-steganalysis ability.

Table 3: Experimental results of anti-steganalysis. The best is highlighted in bold

Dataset	bpw	Methods	ACC ↓	F1 ↓	Pr ↓	Re↓
Quora	0.0821	TS-RNN	0.5762	0.5438	0.5888	**0.5051**
		LS-CNN	0.5614	0.5476	0.5655	**0.5308**
MSCOCO	0.0798	TS-RNN	0.6775	0.5996	0.7905	**0.4830**
		LS-CNN	0.6475	0.6399	0.6540	**0.6265**

(4) Robustness Analysis

As shown in Fig. 2a,b, our method can still successfully extract more than 70% of the secret information in facing three types of attacks. Among them, the extraction success rate even reached 87% in facing replace attacks, and compared with VLC-3 and FLC-3, the advantages of our method are more evident in facing delete attacks.



(a) Quora **(b)** MSCOCO

Figure 2: Robustness test results on the different dataset

To further illustrate the impact of different attack methods, Table 4 presents some examples. Among them, on the Quora dataset, the attribute value of the cover sentence is 0, and the stego sentence with the

attribute value of 1 is obtained by using the adjuster to paraphrase the cover sentence to embed bit 1. On the MSCOCO dataset, the attribute value of the cover sentence is 0, and to embed bit 0, use the cover sentence as the stego sentence. From Table 4, it can be noticed that our method can maintain consistency with the attribute of stego sentences when facing three types of attack methods, further showing that our method has better robustness.

Table 4: Examples of robustness test. The attribute value of the sentence is highlighted in red font

Methods	Quora	MSCOCO
Cover	What makes it easy for polyglots to learn multiple languages? [0]	Two sports players going after the same yellow and purple ball. [0]
Stego	What is the best way to learn multiple languages? [1]	Two sports players going after the same yellow and purple ball. [0]
Substitution	What is the best method to learn multiple languages? [1]	two sports players going after the same xanthous and purple ball. [0]
Delete	What is best way to learn multiple languages? [1]	Two players going after the same yellow and purple ball. [0]
Insert	What is the best way to learn multiple languages to? [1]	Two sports players going after the sports same yellow and purple ball. [0]

5 Conclusion

We proposed a novel linguistic steganography based on sentence attribute encoding. This method solves the problem of traditional modification-based linguistic steganography relies on cumbersome manual features by using a semantic attribute adjuster based on the paraphrase generation model. Moreover, we solve the dependence problem for the generation model during the extraction process by designing a simple semantic attribute analyzer as a proxy model. The experimental results show that our method can greatly improve the extraction efficiency and reduce the consumption of computing resources. Furthermore, the stego texts generated by our method have higher text quality, better anti-steganalysis ability, and also better robustness to three common attack methods. In the future, we will consider exploring the multi-attribute coding method of sentences to improve the embedding capacity and enhance the practicability in the actual scene.

Acknowledgement: The authors gratefully acknowledge the helpful comments and suggestions of the reviewers and editors, which have improved the presentation.

Funding Statement: This project is supported by the National Natural Science Foundation of China under Grant 61972057; and Hunan Provincial Natural Science Foundation of China under Grant 2022JJ30623.

Author Contributions: The authors confirm contribution to the paper as follows: Lingyun Xiang: Conceptualization, Methodology, Writing—original draft, Investigation. Xu He: Methodology, Software, Writing—reviewing & editing. Xi Zhang: Writing—reviewing & editing, Analysis and interpretation of results. Chengfu Ou: Writing—original draft, Visualization, Conceptualization. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All relevant data are within the paper. The data are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Simmons GJ. The prisoners' problem and the subliminal channel. In: Advances in cryptology. New York, NY, USA: Springer; 1984. p. 51–67. doi:10.1007/978-1-4684-4730-9_5.
2. Gao S, Zhang Z, Iu HH-C, Ding S, Mou J, Erkan U, et al. A parallel color image encryption algorithm based on a 2D logistic-rulkov neuron map. IEEE Internet Things J. 2025;12(11):18115–24. doi:10.1109/jiot.2025.3540097.
3. Gao S, Liu J, Iu HHC, Erkan U, Zhou S, Wu R, et al. Development of a video encryption algorithm for critical areas using 2D extended Schaffer function map and neural networks. Appl Math Model. 2024;134(1):520–37. doi:10.1016/j.apm.2024.06.016.
4. Zhang R, Liu J, Zhang R. Controllable semantic linguistic steganography via summarization generation. In: ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2024 Apr 14–19; Seoul, Republic of Korea. p. 4560–4.
5. Wang C, Liu Y, Tong Y, Wang J. GAN-GLS: generative lyric steganography based on generative adversarial networks. Comput Mater Contin. 2021;69(1):1375–90. doi:10.32604/cmc.2021.017950.
6. Wu Y, Yu P, Yuan C. A survey on neural network-based image data hiding for secure communication. Int J Auton Adapt Commun Syst. 2023;16(5):476–93. doi:10.1504/ijaacs.2023.134095.
7. Chen Z, Qin J. Reversible data hiding in encrypted images based on histogram shifting and prediction error block coding. Int J Auton Adapt Commun Syst. 2025;18(1):45–66. doi:10.1504/ijaacs.2025.10061182.
8. Solak S, Abdirashid AM, Adjevi A, Sahu AK. Robust data hiding method based on frequency coefficient variance in repetitive compression. Eng Sci Technol Int J. 2024;56(11):101756. doi:10.1016/j.jestch.2024.101756.
9. Su W, Ni J, Hu X, Li B. Efficient audio steganography using generalized audio intrinsic energy with micro-amplitude modification suppression. IEEE Trans Inf Forensics Secur. 2024;19:6559–72. doi:10.1109/tifs.2024.3417268.
10. Li J, He P, Liu J, Luo J, Xia Q. Cover enhancement method for audio steganography based on universal adversarial perturbations with sample diversification. Comput Mater Contin. 2023;75(3):4893–915. doi:10.32604/cmc.2023.036819.
11. Mao X, Hu X, Peng W, Gan Z, Qian Z, Zhang X, et al. From covert hiding to visual editing: robust generative video steganography. In: Proceedings of the 32nd ACM International Conference on Multimedia. MM '24; 2024 Oct 28–Nov 1; Melbourne, VIC, Australia. p. 2757–65.
12. Li R, Qin J, Tan Y, Xiong NN. Coverless video steganography based on frame sequence perceptual distance mapping. Comput Mater Contin. 2022;73(1):1571–83. doi:10.32604/cmc.2022.029378.
13. Wu J, Wu Z, Xue Y, Wen J, Peng W. Generative text steganography with large language model. In: Proceedings of the 32nd ACM International Conference on Multimedia; 2024 Oct 28–Nov 1; Melbourne, VIC, Australia. p. 10345–53.
14. Majeed MA, Sulaiman R, Shukur Z, Hasan MK. A review on text steganography techniques. Mathematics. 2021;9(21):2829. doi:10.3390/math9212829.
15. Chang CY, Clark S. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing; 2010 Oct 9–11; Cambridge, MA, USA. p. 1194–203.
16. Yajam HA, Mousavi AS, Amirmazlaghani M. A new linguistic steganography scheme based on lexical substitution. In: 2014 11th International ISC Conference on Information Security and Cryptology; 2014 Sep 3–4; Tehran, Iran. p. 155–60.
17. Li M, Mu K, Zhong P, Wen J, Xue Y. Generating steganographic image description by dynamic synonym substitution. Signal Processing. 2019;164(1):193–201. doi:10.1016/j.sigpro.2019.06.014.
18. Topkara M, Topkara U, Atallah MJ. Words are not enough: sentence level natural language watermarking. In: Proceedings of the 4th ACM International Workshop on Contents Protection and Security. MCPS'06; 2006 Oct 28; Santa Barbara, CA, USA. p. 37–46.
19. Meral HM, Sankur B, Özsoy AS, Güngör T, Sevinç E. Natural language watermarking via morphosyntactic alterations. Comput Speech Lang. 2009;23(1):107–25.

20. Murphy B, Vogel C. The syntax of concealment: reliable methods for plain text information hiding. Vol. 6505, In: Security, steganography, and watermarking of multimedia contents IX. Washington, DC, USA: SPIE; 2007. p. 351–62.
21. Chang CY, Clark S. Linguistic steganography using automatically generated paraphrases. In: Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10; 2010; Kerrville, TX, USA: Association for Computational Linguistics. p. 591–9.
22. Wilson A, Ker AD. Avoiding detection on twitter: embedding strategies for linguistic steganography. Electron Imaging. 2016;28(8):1–9. doi:10.2352/issn.2470-1173.2016.8.mwsf-074.
23. Yang ZL, Guo XQ, Chen ZM, Huang YF, Zhang YJ. RNN-stega: linguistic steganography based on recurrent neural networks. IEEE Trans Inf Foren Sec. 2018;14(5):1280–95. doi:10.1109/tifs.2018.2871746.
24. Ziegler Z, Deng Y, Rush AM. Neural linguistic steganography. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. p. 1210–5.
25. Yang ZL, Zhang SY, Hu YT, Hu ZW, Huang YF. VAE-stega: linguistic steganography based on variational auto-encoder. IEEE Trans Inf Foren Sec. 2020;16:880–95. doi:10.1109/tifs.2020.3023279.
26. Zhou X, Peng W, Yang B, Wen J, Xue Y, Zhong P. Linguistic steganography based on adaptive probability distribution. IEEE Trans Dependable Secure Comput. 2021;19(5):2982–97. doi:10.1109/tdsc.2021.3079957.
27. Wang R, Xiang L, Liu Y, Yang C. PNG-stega: progressive non-autoregressive generative linguistic steganography. IEEE Signal Process Lett. 2023;30:528–32. doi:10.1109/lsp.2023.3272798.
28. Yang Z, Xiang L, Zhang S, Sun X, Huang Y. Linguistic generative steganography with enhanced cognitive-imperceptibility. IEEE Signal Process Lett. 2021;28:409–13. doi:10.1109/lsp.2021.3058889.
29. Li Y, Zhang R, Liu J, Lei Q. A semantic controllable long text steganography framework based on LLM prompt engineering and knowledge graph. IEEE Signal Process Lett. 2024;31(4):2610–4. doi:10.1109/lsp.2024.3456636.
30. Long Y, Yang Z, Wang Z, Zhou Z, Huang Y, Zhou L. SCF-stega: controllable linguistic steganography based on semantic communications framework. In: ICASSP 2025–2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2025 Apr 6–11; Hyderabad, India. p. 1–5.
31. Bolshakov IA, Gelbukh A. Synonymous paraphrasing using wordnet and internet. In: Natural language processing and information systems. Berlin/Heidelberg, Germany: Springer; 2004. p. 312–23. doi:10.1007/978-3-540-27779-8_27.
32. Xiang L, Wang X, Yang C, Liu P. A novel linguistic steganography based on synonym run-length encoding. IEICE Trans Inf Syst. 2017;100(2):313–22. doi:10.1587/transinf.2016edp7358.
33. Xiang L, Wu W, Xu L, Yang C. A linguistic steganography based on word indexing compression and candidate selection. Multimed Tools Appl. 2018;77(21):28969–89. doi:10.1007/s11042-018-6072-8.
34. Mahato S, Khan DA, Yadav DK. A modified approach to data hiding in Microsoft Word documents by change-tracking technique. J King Saud Univ-Comput Inf Sci. 2020;32(2):216–24. doi:10.1016/j.jksuci.2017.08.004.
35. Chang CY, Clark S. The secret's in the word order: Text-to-text generation for linguistic steganography. In: Proceedings of COLING 2012; 2012 Dec 8–15; Mumbai, India. p. 511–28.
36. He L, Gui X, Wu R, Xie B, Hu C. A hybrid natural language information hiding system. Elektron Elektrotech. 2012;18(9):95–100. doi:10.5755/j01.eee.18.9.2817.
37. Kim MY, Goebel R. Adaptive-capacity and robust natural language watermarking for agglutinative languages. Secur Commun Netw. 2012;5(3):301–10. doi:10.1002/sec.336.
38. Stutsman R, Grothoff C, Atallah M, Grothoff K. Lost in just the translation. In: SAC '06: Proceedings of the 2006 ACM Symposium on Applied Computing; 2006; New York, NY, USA: Association for Computing Machinery. p. 338–45. doi:10.1145/1141277.1141358.
39. Meng P, Shi YQ, Huang L, Chen Z, Yang W, Desoky A. LinL: Lost in n-best list. In: Information Hiding: 13th International Conference; 2011; Berlin/Heidelberg: Springer. p. 329–41.
40. Yang T, Wu H, Yi B, Feng G, Zhang X. Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding. IEEE Trans Dependable Secure Comput. 2023;21(1):139–52. doi:10.1109/tdsc.2023.3247493.

41. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision; 2015 Dec 7–13; Santiago, Chile. p. 1026–34.
42. Chen X, Fang H, Lin TY, Vedantam R, Gupta S, Dollár P, et al. Microsoft coco captions: data collection and evaluation server. arXiv:1504.00325. 2015.
43. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781. 2013.
44. Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics; 2002 Jul 7–12; Philadelphia, PA, USA. p. 311–8.
45. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: evaluating text generation with BERT. In: 8th International Conference on Learning Representations, ICLR 2020; 2020 Apr 26–30; Ethiopia: Addis Ababa. p. 1–43.
46. Yang Z, Wang K, Li J, Huang Y, Zhang YJ. TS-RNN: text steganalysis based on recurrent neural networks. IEEE Signal Process Lett. 2019;26(12):1743–7. doi:10.1109/lsp.2019.2920452.
47. Wen J, Zhou X, Zhong P, Xue Y. Convolutional neural network based text steganalysis. IEEE Signal Process Lett. 2019;26(3):460–4. doi:10.1109/lsp.2019.2895286.