Hi-Stega: A Hierarchical Linguistic Steganography Framework Combining Retrieval and Generation

Huili Wang[1](✉), Zhongliang Yang[2], Jinshuai Yang[3], Yue Gao[3], and Yongfeng Huang[3,4]

[1] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China
whl21@mails.tsinghua.edu.cn
[2] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
yangzl@bupt.edu.cn
[3] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[4] Zhongguancun Laboratory, Beijing 100094, China

Abstract. Due to the widespread use of social media, linguistic steganography which embeds secret message into normal text to protect the security and privacy of secret message, has been widely studied and applied. However, existing linguistic steganography methods ignore the correlation between social network texts, resulting in steganographic texts that are isolated units and prone to breakdowns in cognitive-imperceptibility. Moreover, the embedding capacity of text is also limited due to the fragmented nature of social network text. In this paper, in order to make the practical application of linguistic steganography in social network environment, we design a hierarchical linguistic steganography (**Hi-Stega**) framework. Combining the benefits of retrieval and generation steganography method, we divide the secret message into data information and control information by taking advantage of the fact that social network contexts are associative. The data information is obtained by retrieving the secret message in normal network text corpus and the control information is embedded in the process of comment or reply text generation. The experimental results demonstrate that the proposed approach achieves higher embedding payload while the imperceptibility and security can also be guaranteed. (All datasets and codes used in this paper are released at https://github.com/wanghl21/Hi-Stega.)

Keywords: Linguistic steganography · Text Generation · Information Security

1 Introduction

With the advancement of technology, social network has gradually become an integral part of people's lives and people pay increasing attention on their privacy. The technology with a long history named steganography [2] is designed

B. Luo et al. (Eds.): ICONIP 2023, LNCS 14451, pp. 41–54, 2024.
https://doi.org/10.1007/978-981-99-8073-4_4

to safeguard the privacy and security of secret message by embedding the message into normal carriers thereby making it difficult for monitors to detect the transmission of secret message.

According to the International Data Corporation (IDC) [18], the global data circle will expand to 163 ZB (1 ZB equals 1 trillion GB) by 2025, which is approximately ten times the data generated in 2016. It is worth mentioning that steganography can be employed with various forms of carriers including image [8], text [11], audio [1], video [12] and so on. The vast volume of data available on social media platforms provides numerous carrier options and scenarios for transmitting secret messages using steganography. However, affected by the transmission channel, the carriers of some media forms like image, audio, and video may be damaged during the transmission process, resulting in a decrease in the robustness of secret information. Among the various media carriers, text steganography is less affected by transmission channels, exhibiting higher levels of robustness and practicality, thus rendering it particularly suitable for social network scenarios.

The linguistic steganography technique based on text context can be categorized into three primary methods: modification-based method, retrieval-based method, and generation-based method. Modification-based methods [3,19,20] embed information by lexically modifying the text content. These methods generally have a lower embedding rate and tend to alter word frequencies, which can make them more susceptible to steganalysis techniques. Retrieval-based methods [4,21,22,30] first encode the samples in the text corpus, and then select the corresponding sentence for transmission by mapping sample with secret message. These methods need to encode and construct the text corpus, and their capacity is relatively low. However, since the text carrier itself has not changed, it is difficult for the retrieval-based method to be detected by the existing steganalysis methods. In recent years, with the development of Natural Language Processing (NLP) technology, generation-based methods [9,10,24,26,27,33] mainly embed secret messages by encoding the conditional probability distribution of each word reasonably in the process of text generation using neural networks. These methods can improve the embedding payload to some extent, but the generated text is semantically random resulting in ineffective resistance to some steganalysis methods [23,28]. The flexibility of generation-based methods, including the ability to adapt to any context and any scene, has not been fully explored and utilized.

There are two significant challenges associated with the application of linguistic steganography methods in social network scenarios. Firstly, in the social network environment, information carriers are no longer isolated semantic expression units. Instead, they are interconnected through social and cognitive relationships. For example, two parties communicate in social network as shown in Fig. 1. Alice says “Hi, long time no see!”. Bob will probably reply that “Yes, how’s it going?”. It will not be that “Eve’s experiment isn’t done.”. Previous text steganography algorithms paid less attention on the relationship between texts in social networks, making them inconsistent in contextual semantics, which may

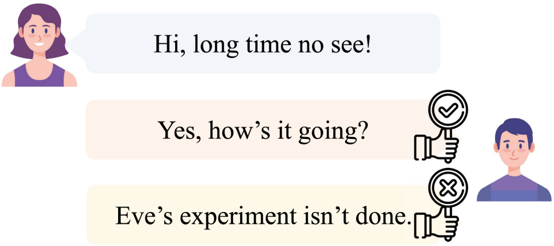


Fig. 1. Two parties communicate in social network.

bring potential security risks. Secondly, the fragmentation of social networks is serious, and we need steganography with higher embedding payload in order to embed certain secret message in short sentences and avoid the security risks associated with continuous communication.

According to the contextual relevance of social network text and the demand for high embedding payload, we propose a new **hierarchical linguistic steganography (Hi-Stega)** framework that combines the strengths of strong imperceptibility of retrieval method and large capacity of generation method. The secret message embedding process is divided into data information process layer and control information embedding layer according to different processing phases. The **data information** is the context data carrier obtained by retrieving the normal social network text (e.g. news) according to the secret message, while the **control information** is the identification signal of the secret message in the context data carrier and is embedded in the process of comment text generation. The proposed framework has the following contributions to application of linguistic steganography in social network. Firstly, it can strengthen the semantic coherence between steganographic text (stegotext) and normal text to improve cognitive imperceptibility [29]. Secondly, the data layer and the control layer are separated, and secret message cannot be decoded only by obtaining a single layer of information, which improves the security of secret message. Thirdly, experimental results demonstrate that by properly designing the language model (LM), the proposed framework has a large embedding payload while the imperceptibility and security can also be guaranteed.

2 Related Works

Steganography is the practice of hiding secret or sensitive information within an ordinary-looking file or message without arousing suspicion. Retrieval-based methodologies initially encode the samples within the textual corpus, subsequently selecting the suitable sentence for transmission by associating the samples with the concealed message through mapping [4,21,22,30]. For example, Chen *et al.* [4] proposed an efficient method of coverless text steganography based on the Chinese mathematical expression. By exploiting the space mapping concept, Wang *et al.* [21] initialized a binary search tree based on the texts from the Internet and searched the corresponding texts according to the secret binary digit string generated from a secret. These methods have the advantage of leveraging existing text resources, which can make the hidden message less conspicuous. However, the payload and effectiveness of retrieval-based steganography heavily depends on the quality and size of the underlying dataset.

The advancement of NLP technology has facilitated the development of generation-based linguistic steganography methods that primarily utilize neural networks like Recurrent Neural Network (RNN) [26], variational autoencoder (VAE) [27] and transformers [9,10,24,33] to learn the statistical language model and then employ the encoding algorithm to encode the conditional probability distribution of each word in the generation process to embed secret information. These approaches offer several advantages, including improved quality of steganographic texts and higher embedding payload. Besides, some linguistic steganography algorithms are concerned with the semantics of generating text [9,24,31]. For instance, Zhang *et al.* [31] introduced a generation-based linguistic steganography method that operates in the latent space, encoding secret messages into implicit attributes (semantemes) present in natural language. Yang *et al.* [24] involved pivoting the text between two distinct languages and embedded secret data using a strategy that incorporated semantic awareness into the information encoding process. Nevertheless, these approaches overlook the semantic relationship between sentence contexts, rendering them vulnerable to some steganalysis methods [23,28]. Consequently, this paper aims to address these limitations and proposes an improved method combined advantages both of retrieval-based and generation-based methods.

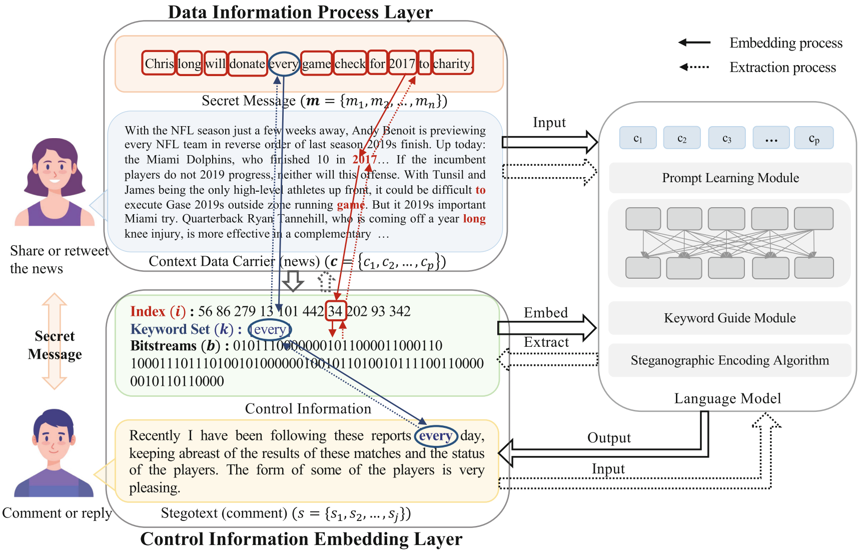
3 The Proposed Framework

As analyzed above, linguistic steganography in the social network environment should not only satisfy perceptual and statistical imperceptibility, but also make stegotext and context semantics coherent, and expand the steganographic embedding payload as much as possible. The proposed framework combines retrieval and generation of text and focuses on addressing these challenges to make linguistic steganography practical and applicable to social network scenarios.

The core idea is to retrieve the appropriate context data carrier $\boldsymbol{c}$ in the network social text corpus C according to the secret message $\boldsymbol{m}$ and then we can perform information embedding when we make comments or replies on it. Therefore, the linguistic steganography adapted to the social network can be expressed as:

$$\begin{aligned} &Embed(\boldsymbol{m}, \boldsymbol{c}) \rightarrow \boldsymbol{s}, \\ &\text{s.t.} \begin{cases} \min D_{SD}(P_S || P_C) \\ \mid \varDelta D_{SC}(\cdot) \mid = \mid D_{SC}(\boldsymbol{s}, \boldsymbol{c}) - D_{SC}(\boldsymbol{c}, \boldsymbol{c}) \mid \leq \varepsilon (\varepsilon \rightarrow 0), \end{cases} \end{aligned} \tag{1}$$

where $Embed(\cdot)$ is the overall description of the Hi-Stega, and $\boldsymbol{s}$ is the stegotext and also the comment to the context $\boldsymbol{c}$. $D_{SD}(\cdot)$ is implemented to measure


Data Information Process Layer
Chris long will donate every game check for 2017 to charity.
Secret Message ($\boldsymbol{m} = \{m_1, m_2, \dots, m_n\}$)
Context Data Carrier (news) ($\boldsymbol{c} = \{c_1, c_2, \dots, c_p\}$)
Share or retweet the news
Secret Message
Comment or reply
Index ($\boldsymbol{i}$) : 56 86 279 13 101 442 34 202 93 342
Keyword Set ($\boldsymbol{k}$) : every
Bitstreams ($\boldsymbol{b}$) : 0101110000000101100001100011010001110111010010100000010010110100101111001100000010110110000
Control Information
Stegotext (comment) ($s = \{s_1, s_2, \dots, s_j\}$)
Control Information Embedding Layer
Embedding process
Extraction process
Input
Embed
Extract
Output
Input
c_1
c_2
c_3
…
c_p
Prompt Learning Module
Keyword Guide Module
Steganographic Encoding Algorithm
Language Model

With the NFL season just a few weeks away, Andy Benoit is previewing every NFL team in reverse order of last season 2019s finish. Up today: the Miami Dolphins, who finished 10 in **2017**… If the incumbent players do not 2019 progress, neither will this offense. With Tunsil and James being the only high-level athletes up front, it could be difficult **to** execute Gase 2019s outside zone running **game**. But it 2019s important Miami try. Quarterback Ryan Tannehill, who is coming off a year **long** knee injury, is more effective in a complementary …

Recently I have been following these reports **every** day, keeping abreast of the results of these matches and the status of the players. The form of some of the players is very pleasing.

Fig. 2. The proposed Hi-Stega framework.

differences between the statistical distribution of the stegotext P_S and normal text P_C. $D_{SC}(\cdot)$ is the function that measures the degree of contextual semantic coherence. Based on an example of the steganographic process, the overall design of the proposed framework is shown in Fig. 2 and the pseudocode of the embedding process of the proposed framework is demonstrated in Algorithm 1. We divide the secret message embedding process into data information process layer and control information embedding layer. It is reasonable to treat social networks as a huge real-time updated corpus C. The main task for the upper layer is to retrieve secret message $\boldsymbol{m}$ in the social network environment to find the appropriate context data carrier $\boldsymbol{m}$ (ensure that as many words in the secret message as possible appear in the context data carrier). The secret message and the context data carrier are processed to obtain control information which includes the indexes $\boldsymbol{i}$ (the positions of the words in the secret message appear in the context data carrier) and the words $\boldsymbol{k}$ (named keyword set) that do not appear. In the lower layer, the information returned from the upper layer needs to be processed and transformed into bit stream $\boldsymbol{b}$ through a format protocol. Bit stream embedding is performed during the process of generating text with the context data carrier as model input. It is worth noting that the process of generation is guided using $\boldsymbol{k}$ to ensure that some of the words in the secret message which may not be covered by the context data carrier will appear in the generated stegotext $\boldsymbol{s}$. At the same time, the semantics of the generated text is directed in the relevant direction and semantic coherence is enhanced.

Algorithm 1. Secret Message Embedding Algorithm.

```
Input: Secret message $\boldsymbol{m}$, social network text corpus $C$, Language Model $LM(\cdot)$.
Output: Stegotext $\boldsymbol{s}$, steganography parameters $\Phi$.
function DataInformationProcessing($\boldsymbol{m}, C$)
    Retrieve in $C$ to find $\boldsymbol{c}$ can that cover as many words in $\boldsymbol{m}$ as possible;
    Record the indexes $\boldsymbol{i} = \{i_1, i_2, \ldots, i_p\}$ of these words in $\boldsymbol{c}$;
    if $m_i \in \boldsymbol{m}$ and $m_i \notin \boldsymbol{c}$ then
        Add $m_i$ to keywords set $\boldsymbol{k}$;
    end if
    return Control information $\boldsymbol{k}$, $\boldsymbol{i}$, and data carrier $\mathbf{c}$;
end function
function Control Information Embedding($\boldsymbol{k}, \boldsymbol{i}, \boldsymbol{c}$)
    Convert $\boldsymbol{i}$ into bit stream $\boldsymbol{b} = \{b_1, b_2, \ldots, b_l\}$ through a format protocol;
    while Generation process is not end do
        $w_j \leftarrow LM(\boldsymbol{c}, \boldsymbol{k}, \boldsymbol{b})$;
        if $w_j \in \boldsymbol{k}$ then
            Remove $w_j$ from $\boldsymbol{k}$;
            Add index $j$ into parameters set $\Phi$;
        end if
    end while
    return Stegotext $\boldsymbol{s} = \{w_1, w_2, \ldots, w_s\}$ and $\Phi$.
end function
$(\boldsymbol{k}, \boldsymbol{i}, \boldsymbol{c}) \leftarrow$ DataInformationProcessing($\boldsymbol{m}, C$);
$\boldsymbol{s}, \Phi \leftarrow$ ControlInformationEmbedding($\boldsymbol{k}, \boldsymbol{i}, \boldsymbol{c}$);
Negotiate parameters with the receiver, such as comment time means $\Phi$, etc.
```

The extraction algorithm is the reverse process of the embedding algorithm, where the control information is obtained and then mapped to the context data carrier to recover the secret message. The pseudocode is demonstrated in Algorithm 2.

4 Methodology of Our Implementation

To verify the effectiveness of the proposed Hi-Stega framework, we construct a concrete model for further assessment.

4.1 Steganographic Text Generation

As shown in the right half of Fig. 2, our steganographic text is finally obtained by using a generative steganography algorithm, so in the first step we describe its overall process. Text can be regarded as a token sequence, composed of specific words according to the semantic association and syntactic rules. In Hi-Stega, the context data carrier $\boldsymbol{c}$ retrieved according to the secret message $\boldsymbol{m}$ can be regarded as the input of the language model and the chain rule is used to describe

Algorithm 2. Secret Message Extraction Algorithm.

```
Input: Context $\boldsymbol{c}$, steganography parameters $\Phi$, stegotext $\boldsymbol{s}$, Language Model $LM(\cdot)$.
Output: Secret message $\boldsymbol{m} = \{m_1, m_2, m_3, \ldots, m_n\}$.
Input $\boldsymbol{c}$ and prompt phase to $LM(\cdot)$.
if $\Phi == \phi$ then
    Decode algorithm outputs embedded bits $\boldsymbol{b}$;
else
    Recover keywords set $\boldsymbol{k}$ from $\boldsymbol{c}$ and $\Phi$;
    Decode algorithm outputs embedded bits $\boldsymbol{b}$ with $\boldsymbol{k}$;
end if
$\boldsymbol{m} \leftarrow (\boldsymbol{i}) \leftarrow \boldsymbol{b}$;
```

the probability distribution of word sequences:

$$p_\theta(y) = \prod_{t=1}^{|y|} p_\theta(y_t \mid (\boldsymbol{c}, y_{<t})), \tag{2}$$

where the distributional probability p_θ is typically parameterized by a neural network with parameters θ, e.g., a transformer [5,14,17]. Generative steganography algorithm mainly embeds secret information by adjusting and encoding the probability distributions p_θ of candidate words.

4.2 Prompt Learning Module (PL Module)

After inputting context data carrier $\boldsymbol{c}$ into the language model, how to make the generated stegotext $\boldsymbol{s}$ semantically coherent like the normal social text is the problem that proposed Hi-Stega needs to pay attention to. We adopt the prompt learning [7] by adding some specific words as prompts after the input, which can guide LM to generate better sentence vectors and improve the consistency of contextual semantics. According to the different social scenarios, different prompt templates can be used. For example, here we exploit the correlation between news and comments for Hi-Stega, the template can be "[$\boldsymbol{c}$] comment is [$mask$]". Given the data context carrier $\boldsymbol{c}$, we can map $\boldsymbol{c}$ to $\boldsymbol{c}_{prompt}$ with the template, then generate better sentence representation.

4.3 Keyword Guide Module (KG Module)

As introduced in Algorithm 1, sometimes the secret message $\boldsymbol{m}$ cannot be completely covered in the process of retrieving the context data carrier $\boldsymbol{c}$. At this time, it is necessary to include the uncovered secret words in the stegotext generation process. The keyword guide module is used to adjust the p_θ in Eq. 2 to make the semantics of the generated text closer to the keywords set $\boldsymbol{k} = \{k_1, k_2, \ldots, k_n\}$. Here, inspired by [15], we make a simple modification to the distributional probability $p_\theta(\cdot)$ to guide generation towards keyword $k_i \in \boldsymbol{k}$:

$$p_{\theta}^{'}(y_t, k_i \mid y_{<t}) = p_{\theta}(y_t \mid y_{<t}) + \alpha \cdot \frac{1 + D_{SC}(y_t, k_i)}{2}. \tag{3}$$

Here we use the $D_{SC}(y_t, k_i) = cosine_{similarity}(glove(y_t), glove(k_i))$[1]. As explained in [15], we can control the parameters α in Eq. 3 to guarantee the appearance of specified keyword k_i in the generated sentences. For instance, we can increase α on an exponential schedule which means that as the generated sequence increases in length, so does the strength of the semantic shift, until we deterministically choose the guide keyword k_i. The parameters α at time t can be expressed:

$$\alpha_t = \begin{cases} \alpha_0 exp\{\frac{t-t_j}{T-|\boldsymbol{k}|-t_j}\}, & \textbf{if } t < T- \mid \boldsymbol{k} \mid \\ \infty, & \textbf{otherwise} \end{cases} \tag{4}$$

where T is the maximum length of the sentence, and previous keyword appeared at t_j.

5 Experiments and Analysis

5.1 Dataset

To simulate the real social network environment, we adopted a natural corpus including news and its comments from Yahoo! News [25]. Here we preprocessed the raw data and calculated the basic statistical information. On average news titles, bodies, and comments contain 12, 498, and 32 words respectively. The titles of news generally introduce the main information such as time, place, people, and events clearly and concisely, which have similarity with the intelligence message delivered. Therefore, we treated news titles as secret messages to be delivered in our experiments. In addition to the comments themselves, the dataset includes the number of upvotes and downvotes. It can be speculated that the comments with the maximum sum of upvotes and downvotes which can stimulate people's discussion are more in line with the comment relationship of news in the social network environment. We calculated the semantic coherence metric between the two as the baseline.

5.2 Experimental Setting

We employed the GPT-2 [17] as our pre-training language model. Based on this, comparison experiments are conducted by different fine-tuning methods as well as information embedding methods.

- GPT-2: Without fine-tuning GPT-2, random bit streams are embedded during the generation.
- $model^{*}_{base}$: GPT-2 is fine-tuned with above dataset and random bit streams are embedded during the generation.

[1] Glove for Word Representation in https://github.com/stanfordnlp/GloVe.

- $model_{base}$: GPT-2 is fine-tuned with above dataset with PL and random bit streams are embedded during the generation.
- Hi^{*}_{Stega}: GPT-2 is fine-tuned with above dataset and the secret message is embedded according to the process of Algorithm 1.
- Hi_{Stega}: GPT-2 is fine-tuned with above dataset with PL and the secret message is embedded according to the process of Algorithm 1.

Specifically, when fine-tuning the pre-trained language model, we used AdamW optimizer [13] with an initial learning rate which was 8e−4 and a linear scheduler type. The parameters α_0 was set to 5.0. The number of training epochs was set to 100.

To fully verify the effectiveness of the proposed framework, we employed three commonly used steganographic encoding algorithms including Huffman Coding (HC) [26], Arithmetic Coding (AC) [33], and Adaptive Dynamic Grouping (ADG) [32]. We tried to retrieve the news titles (2,400 pieces) in the news dataset (4,500 pieces) (news corresponding to the above titles is not included) and generated stegotext according to Algorithm 1. For statistical and perceptual imperceptibility evaluation, we calculated the average *perplexity*(*ppl*), *distinct*-*n* and *mauve* [16]. For cognitive imperceptibility evaluation, we calculated two standard metrics including $\Delta(cosine)$[2] and $\Delta(simcse)$[3] [6], both of which measure the gap of the semantic coherence between the stegotext and the context data carrier and the normal news-comment semantic coherence. At the same time, we defined two types of embedding payload: practice embedding payload ER_1 which means how many bits per word are embedded on average in the text generation process and effective embedding payload ER_2 which computes the average equivalent information bits embedded per word contained in the stegotext under the Hi-Stega framework:

$$\begin{cases} ER_1 = \frac{|b|}{|\boldsymbol{s}|} & \text{if } \boldsymbol{s} \leftarrow Embed(b), \\ ER_2 = \frac{8\cdot\sum_0^n |m_i|}{|\boldsymbol{s}|} & \text{if } \boldsymbol{s} \leftarrow Embed(\boldsymbol{m}, \boldsymbol{c}) \end{cases} \tag{5}$$

where a letter needs 8 bits to be represented and b is the bit stream embedded in the language model generation process.

5.3 Imperceptibility Analysis

Metrics including text fluency and diversity are recorded in Table 1, and these values reflect the perceptual and cognitive imperceptibility of the stegotext generated by different methods. Firstly, the *ppl* values of Hi_{Stega} is much smaller than other models, indicating that the proposed Hi_{Stega}, which only embeds control information, can generate higher quality text than the directly embedding secret information bit stream methods. Secondly, the *mauve* value of our method is much higher than that of the comparison methods, even about 10

[2] It is the cosine distance of the sentence embedding using https://huggingface.co/sentence-transformers.

[3] https://github.com/princeton-nlp/SimCSE.

Table 1. Comparison of fluency and diversity coherence of steganographic texts generated by different methods.

Algorithm	Model	Fluency		Diversity		
		ppl (↓)	*mauve*(↑)	$distinct_2$	$distinct_3$	$distinct_4$
HC [26]	GPT-2	213.94	0.0135	0.925	0.976	0.984
	$model^*_{base}$	152.64	0.0712	**0.961**	**0.991**	0.996
	$model_{base}$	211.50	0.0403	0.959	0.990	**0.998**
	Hi^*_{Stega}	112.34	0.1068	0.856	0.905	0.925
	Hi_{Stega}	**109.60**	**0.1341**	0.869	0.920	0.938
AC [33]	GPT-2	251.53	0.0184	0.960	0.979	0.982
	$model^*_{base}$	279.90	0.0668	**0.967**	**0.991**	**0.998**
	$model_{base}$	290.34	0.0425	0.964	0.987	0.994
	Hi^*_{Stega}	182.44	0.1499	0.853	0.895	0.914
	Hi_{Stega}	**174.44**	**0.2051**	0.870	0.909	0.926
ADG [32]	GPT-2	395.11	0.0175	0.946	0.971	0.975
	$model^*_{base}$	287.04	0.1253	**0.966**	**0.990**	**0.995**
	$model_{base}$	259.88	0.0534	0.959	0.985	0.992
	Hi^*_{Stega}	147.70	0.0937	0.860	0.905	0.924
	Hi_{Stega}	**143.77**	**0.1413**	0.864	0.911	0.930

times that of the comparison method GPT-2, indicating that there is little difference between the distribution of stegotext and the distribution of normal text. Thirdly, the diversity of the text generated by the proposed method is slightly lower due to the KG module, if we can expand the scope of text retrieval and find a more suitable context data carrier, we can further reduce the semantic deviation and statistical bias brought by the KG module. In addition, it can be seen that the results are further improved by adding the PL module to the model, indicating that the PL module is of enhanced significance for the generation of contextual text.

The results of embedding payload and contextual semantic coherence are recorded in Table 2. While the embedding payload ER_1 of the comparison methods including GPT-2, $model^*_{base}$ and $model_{base}$ is larger, such large embedding capacity leads to semantic consistency deviation of context. And the more bits embedded in the text generation process, the worse the quality of the generated text. The embedding payload ER_1 of our method is small, but equivalent effective information embedding payload ER_2 is twice or more than that of the comparison methods. In addition, the semantic coherent metrics of Hi_{Stega} are the closest to those of the normal yahoo news dataset and the magnitude of the difference is around 10−e2 which can be ignored almost. This indicates that although we embed secret information in our generated comment, the comment itself does not semantically draw the attention of third parties. It remains consistent with the context data carrier, ensuring strong cognitive imperceptibility.

Table 2. Comparison of embedding payload and semantic coherence of steganographic texts generated by different methods.

Algorithm	Model	Payload		Semantic Coherent	
		ER_1	ER_2	$\lvert \Delta(cosine) \rvert$ (↓)	$\lvert \Delta(simcse) \rvert$ (↓)
HC [26]	GPT-2	3.52	–	0.2266	0.3114
	$model^*_{base}$	**3.56**	–	0.2209	0.3755
	$model_{base}$	3.50	–	0.2137	0.3331
	Hi^*_{Stega}	2.15	9.16	0.0680	0.1032
	Hi_{Stega}	2.21	**9.34**	**0.0282**	**0.0321**
AC [33]	GPT-2	**6.26**	–	0.2311	0.3226
	$model^*_{base}$	5.20	–	0.2187	0.2699
	$model_{base}$	4.72	–	0.2100	0.3632
	Hi^*_{Stega}	2.32	9.89	0.0542	0.0889
	Hi_{Stega}	2.46	**10.42**	**0.0088**	**0.0191**
ADG [32]	GPT-2	**4.91**	–	0.2306	0.3066
	$model^*_{base}$	3.88	–	0.2213	0.3700
	$model_{base}$	3.58	–	0.2121	0.3583
	Hi^*_{Stega}	2.12	9.25	0.0593	0.0933
	Hi_{Stega}	2.18	**9.40**	**0.0275**	**0.0376**

5.4 Anti-steganalysis Ability

We utilized two of most commonly used steganalysis methods [23,28] to distinguish stegotext from normal news comment. Figure 3 records the detection accuracy and F1 score of each steganalysis model for different steganographic methods and encoding algorithms. We can find that the stegotext generated by Hi^*_{Stega} and Hi_{Stega} has a stronger ability to resist various steganalysis models when using different encoding algorithms. This shows that the stegotext generated by the proposed method is more consistent with the distribution of normal comments in the social network environment and more secure than the previous methods. In addition, we only detect the stegotext here. Actually, the secret message is scattered in the context data carrier and the generated stegotext, and it is not possible to extract the complete secret message by obtaining only the stegotext itself. Besides, it is more difficult to detect when the two are considered as a whole according to the previous detection methods.

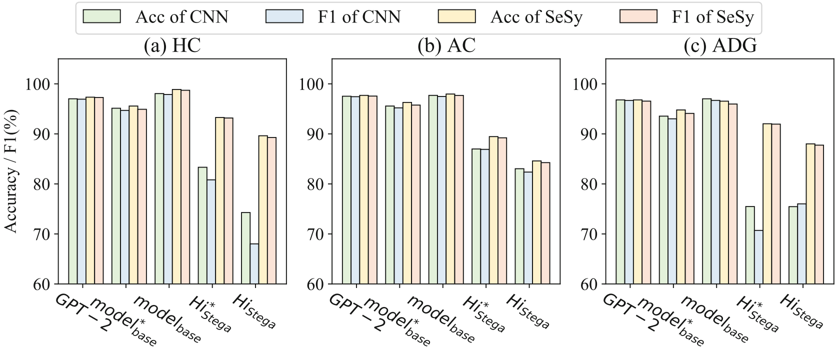


Fig. 3. Performance of different methods when using different steganalysis methods.

6 Conclusion

In order to make linguistic steganography applied in the real social network environment, we design a hierarchical steganography framework that combines retrieval and generation by exploiting the relevance of the social network context. Specific implementation results demonstrate that the proposed framework synthesizes the advantages of both steganography paradigms and enhances the contextual semantic coherence, while also extending the steganographic capacity and improving the resistance to steganalysis. We believe it can be well applied in the social network environment.

References

1. AlSabhany, A.A., Ali, A.H., Ridzuan, F., Azni, A., Mokhtar, M.R.: Digital audio steganography: systematic review, classification, and analysis of the current state of the art. Comput. Sci. Rev. **38**, 100316 (2020)
2. Anderson, R.J., Petitcolas, F.A.: On the limits of steganography. IEEE J. Sel. Areas Commun. **16**(4), 474–481 (1998)
3. Chang, C.Y., Clark, S.: Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. Comput. Linguist. **40**(2), 403–448 (2014)
4. Chen, X., Sun, H., Tobe, Y., Zhou, Z., Sun, X.: Coverless information hiding method based on the Chinese mathematical expression. In: Huang, Z., Sun, X., Luo, J., Wang, J. (eds.) ICCCS 2015. LNCS, vol. 9483, pp. 133–143. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27051-7_12
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910 (2021)
7. Jiang, T., et al.: PromptBERT: improving BERT sentence embeddings with prompts. arXiv preprint arXiv:2201.04337 (2022)

8. Kadhim, I.J., Premaratne, P., Vial, P.J., Halloran, B.: Comprehensive survey of image steganography: techniques, evaluations, and trends in future research. Neurocomputing **335**, 299–326 (2019)
9. Kang, H., Wu, H., Zhang, X.: Generative text steganography based on LSTM network and attention mechanism with keywords. Electron. Imaging **2020**(4), 291-1 (2020)
10. Kaptchuk, G., Jois, T.M., Green, M., Rubin, A.D.: Meteor: cryptographically secure steganography for realistic distributions. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 1529–1548 (2021)
11. Krishnan, R.B., Thandra, P.K., Baba, M.S.: An overview of text steganography. In: 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–6. IEEE (2017)
12. Liu, Y., Liu, S., Wang, Y., Zhao, H., Liu, S.: Video steganography: a review. Neurocomputing **335**, 238–250 (2019)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
14. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 234–239. IEEE (2012)
15. Pascual, D., Egressy, B., Meister, C., Cotterell, R., Wattenhofer, R.: A plug-and-play method for controlled text generation. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3973–3997 (2021)
16. Pillutla, K., et al.: MAUVE: measuring the gap between neural text and human text using divergence frontiers. In: Advances in Neural Information Processing Systems, vol. 34, pp. 4816–4828 (2021)
17. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
18. Reinsel, D., Gantz, J., Rydning, J.: Data age 2025: the evolution of data to life-critical. Don't Focus on Big Data 2 (2017)
19. Wai, E.N.C., Khine, M.A.: Modified linguistic steganography approach by using syntax bank and digital signature. Int. J. Inf. Educ. Technol. **1**(5), 410 (2011)
20. Wang, F., Huang, L., Chen, Z., Yang, W., Miao, H.: A novel text steganography by context-based equivalent substitution. In: 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013), pp. 1–6. IEEE (2013)
21. Wang, K., Gao, Q.: A coverless plain text steganography based on character features. IEEE Access **7**, 95665–95676 (2019)
22. Xiang, L., Wu, W., Li, X., Yang, C.: A linguistic steganography based on word indexing compression and candidate selection. Multimedia Tools Appl. **77**(21), 28969–28989 (2018). https://doi.org/10.1007/s11042-018-6072-8
23. Yang, J., Yang, Z., Zhang, S., Tu, H., Huang, Y.: SeSy: linguistic steganalysis framework integrating semantic and syntactic features. IEEE Sig. Process. Lett. **29**, 31–35 (2021)
24. 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)
25. Yang, Z., Xu, C., Wu, W., Li, Z.: Read, attend and comment: a deep architecture for automatic news comment generation. 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), pp. 5077–5089 (2019)

26. Yang, Z.L., Guo, X.Q., Chen, Z.M., Huang, Y.F., Zhang, Y.J.: RNN-Stega: linguistic steganography based on recurrent neural networks. IEEE Trans. Inf. Forensics Secur. **14**(5), 1280–1295 (2018)
27. Yang, Z.L., Zhang, S.Y., Hu, Y.T., Hu, Z.W., Huang, Y.F.: VAE-Stega: linguistic steganography based on variational auto-encoder. IEEE Trans. Inf. Forensics Secur. **16**, 880–895 (2020)
28. Yang, Z., Wei, N., Sheng, J., Huang, Y., Zhang, Y.J.: TS-CNN: text steganalysis from semantic space based on convolutional neural network. arXiv preprint arXiv:1810.08136 (2018)
29. Yang, Z., Xiang, L., Zhang, S., Sun, X., Huang, Y.: Linguistic generative steganography with enhanced cognitive-imperceptibility. IEEE Sig. Process. Lett. **28**, 409–413 (2021)
30. Zhang, J., Xie, Y., Wang, L., Lin, H.: Coverless text information hiding method using the frequent words distance. In: Sun, X., Chao, H.-C., You, X., Bertino, E. (eds.) ICCCS 2017. LNCS, vol. 10602, pp. 121–132. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68505-2_11
31. Zhang, S., Yang, Z., Yang, J., Huang, Y.: Linguistic steganography: from symbolic space to semantic space. IEEE Sig. Process. Lett. **28**, 11–15 (2020)
32. Zhang, S., Yang, Z., Yang, J., Huang, Y.: Provably secure generative linguistic steganography. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 3046–3055 (2021)
33. Ziegler, Z., Deng, Y., Rush, A.M.: 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), pp. 1210–1215 (2019)