technologies

MDPI

Article

Emotionally Controllable Text Steganography Based on Large Language Model and Named Entity

Hao Shi, Wenpu Guo * and Shaoyuan Gao

College of Communication Engineering, Rocket Force University of Engineering, Xi'an 710025, China; shzxx109829@163.com (H.S.); khalig@foxmail.com (S.G.)
* Correspondence: gwp_403@163.com

Abstract

For the process of covert transmission of text information, in addition to the need to ensure the quality of the text at the same time, it is also necessary to make the text content match the current context. However, the existing text steganography methods excessively pursue the quality of the text, and lack constraints on the content and emotional expression of the generated steganographic text (stegotext). In order to solve this problem, this paper proposes an emotionally controllable text steganography based on large language model and named entity. The large language model is used for text generation to improve the quality of the generated stegotext. The named entity recognition is used to construct an entity extraction module to obtain the current context-centered text and constrain the text generation content. The sentiment analysis method is used to mine the sentiment tendency so that the stegotext contains rich sentiment information and improves its concealment. Through experimental validation on the generic domain movie reviews dataset IMDB, the results prove that the proposed method has significantly improved hiding capacity, perplexity, and security compared with the existing mainstream methods, and the stegotext has a strong connection with the current context.

Keywords: security; information hiding; text steganography; natural language generation; large language model; named entity; sentiment analysis

check for updates

Academic Editors: Fazlullah Khan, Sikha Bagui and Ateeq Ur Rehman

Received: 6 May 2025
Revised: 15 June 2025
Accepted: 19 June 2025
Published: 21 June 2025

Citation: Shi, H.; Guo, W.; Gao, S. Emotionally Controllable Text Steganography Based on Large Language Model and Named Entity. *Technologies* **2025**, *13*, 264. https://doi.org/10.3390/technologies13070264

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous development of network technology and information digitisation, digital communication occupies an important position in daily life, which puts forward higher requirements for information security in cyberspace. In order to protect the medium can be transmitted safely, at first, researchers protect the information by taking encryption. Among the types of media include text [1], pictures [2–4], and video [5]. However, encryption algorithms, while protecting the medium for transmission, expose the encrypted medium to potential attackers and make it vulnerable to targeted attacks [6]. To solve the above problem, steganography has received a lot of attention.

Steganography is a technique of embedding secret information in a digital medium with the aim of achieving covert communication of information [7]. Among them, text, as one of the commonly used carriers in digital communication, has led to numerous cases of malicious theft of text information due to the openness of the network. Text steganography is a text-based information hiding technology, which can realise the covert transmission of secret text in the open channel, and has gradually become a research hotspot in the field of information network security [8]. In recent years, with the development of

natural language processing (NLP) and language models, generative text steganography has received extensive attention [9]. It uses the language model as the text generator, and constructs the correspondence between the predicted word and the secret information bit stream through the encoding strategy, so as to realize the stegotext generation while embedding the secret information.

However, generative text steganography generates stegotext through the probability distribution of the output of the language model [10]. This means that the generation of words at each step is determined by the words nearby the word and the secret information, rather than by the semantics and the current context [11]. If the language model selects a word that violates the context at the current time step, it will lead to a deviation in the semantics of subsequent text generation. The above problems of deviation between stegotext and context content and sentiment affect the imperceptibility of stegotext, making it easier to be detected and difficult to realize covert information transmission.

To solve the above problems, it is important to find a way to increase the semantic and emotional information we can obtain from the context. Therefore, this paper extracts the central text and emotional features in the current context information as the input of LLM to constrain the generation of stegotext, ensure that the text content meets the context requirements, and improve the quality, imperceptibility and security of stegotext.

In summary, the contributions of this work can be outlined as follows.

- Proposes a controllable generative text steganography framework based on LLM. The LLM is used as the stegotext generator without extra training, and the controllable g eneration of stegotext is realized by only using the cue word template, avoiding the complex training process of the traditional deep neural network (DNN) model.
- For the first time, named entity and sentiment analysis are integrated into the process of generating stegotext. By identifying and extracting the central text and emotional features of the current context, the generation of stegotext is constrained. On the basis of information hiding, the consistency between the stegotext and the current context is guaranteed.
- Extensive experiments on different models. Compared with the mainstream advanced algorithms in recent years, the framework proposed in this paper has significantly improved in all aspects, indicating that the framework has better performance in text quality, concealment and security.

The rest of this paper involves the following parts: Section 2 introduces representative text steganography. Section 3 presents the background of text steganography. Section 4 describes the overall framework and design details of the proposed model in this paper. Section 5 is indicated the results and analysis through a series of experiments. Finally, the full text is summarized in Section 6.

2. Related Works

Existing text steganography is mainly divided into three categories [12]: text modification-based steganography, text search-based steganography and text generation-based steganography.

Steganography based on text modification mainly hides secret information by using text features or modifying content, such as modifying syntactic structure [13], synonym substitution [14], and using document redundancy information [15]. The embedding capacity of this method is small, and the security is not enough. Under the attack of specific steganalysis algorithm, it can hardly hide the existence of secret information [16]. Steganography based on text search does not modify the text, and selects the information that meets the requirements as the stegotext by constructing a specific text database [17]. The text generated by this method is directly related to the constructed text database, and

the generated content cannot be transformed with the context. Text generation-based steganography uses generation algorithms to automatically generate stegotext, subject to the constraints of secret information and encoding strategies. This method hides secret information in the process of stegotext generation, makes full use of the redundant space of the text, and improves the hiding capacity [18]. At the same time, with the improvement of the degree of freedom in the process of generating text, the stegotext can be transformed according to the different context, and it is more in line with the natural text [19], which has become the research focus in the field of text steganography.

According to the different generation algorithms, steganography based on text generation can be divided into three categories: rule-based, Markov model and neural network model [20]. In the early steganography of development, researchers mainly designed a rule to generate texts under its constraints [21]. The method is simple to generate, and does not consider the relationship between text sentences, resulting in poor quality of stegotext. In order to solve the problems of single content and insufficient sentence association of stegotext generated by the above methods, Markov model was used as the language model to generate stegotext [22]. However, due to its own limitations, steganography based on Markov model has a poor dependence on long sequences, resulting in poor stegotext quality and difficulty in avoiding the detection of steganalysis algorithms [23].

In recent years, with the continuous development of NLP and deep learning (DL), generative text steganography based on neural network models have been proposed. By using the neural network model as the generation model, the hiding capacity, text quality and security of stegotext have been greatly improved [12]. In this method, the trained language model is used to calculate the candidate words and their corresponding conditional probabilities at each moment, and the stegotext is generated under the constraints of secret information and coding strategy [24].

Fang et al. [19] first used a neural network model as a language generator by using a long short-term memory (LSTM) network as a language model. In the process of candidate word selection, the coding strategy of block coding is used to group the candidate words. Then, the corresponding group was selected under the constraint of the secret information and the word with the maximum conditional probability was selected as the output. Yang et al. [25] used recurrent neural network (RNN) as language models. According to the probability distribution of candidate words, fixed-length coding (FLC) and Huffman coding (HC) are used to encode them, respectively. Finally, the candidate words corresponding to the encoding value are selected according to the secret information until the stegotext is generated. Xiang et al. [26] proposed a character-level steganography based on LSTM. In the process of text generation, word prediction is changed to character prediction, which improves the hiding capacity of stegotext. Yang et al. [27] used a generative adversarial network (GAN) to generate stegotext, and at the same time used the reward function as the loss function of the generator, which solved the problem that GAN was difficult to generate discrete data. Zhou et al. [28] used pseudo-random sampling to obtain the target word and its probability based on GAN. Similar words are obtained through the probability similarity function to form a candidate pool, and then the corresponding candidate words are selected to generate stegotext. In order to improve the language model's understanding of the relationship between sentence order and semantics of text, a text steganography based on encoder-decoder structure is proposed. Yang et al. [29] used an RNN-based decoder-encoder. By combining reinforcement learning in the training process and encoding the candidate words with FLC, the stegotext with better semantics is generated. On this basis, Yang et al. [30] introduced gated recurrent unit and attention mechanism to solve the problem of long-term dependence and improve the fluency and semantic relevance of stegotext. Yang et al. [31] used variational auto-encoder (VAE) as text generation, LSTM

and bidirectional encoder representations from transformers (BERT) as encoders and RNN as decoder, respectively. HC and arithmetic coding (AC) strategies are used to improve the security of the stegotext. To solve the problem that the distribution tends to be diversified due to different sentence semantics, Tu et al. [32] used two independent VAE to model the topic and sequence of the stream model and the discriminator. By learning the latent content of the topic and combining the sequence knowledge and the learned topic content into the text generation process, semantically consistent high-quality text is generated.

In recent years, pre-trained models have been widely used in the field of text steganography because they can generate high-quality, coherent and more contextual text. Ziegler et al. [33] used GPT-2 as the language model, while selecting AC in the encoding strategy, and generated texts that were more in line with the statistical language model. Based on GPT-2, Pang et al. [34] reconstructed softmax by introducing an adjustment factor. It improves the conditional probability of low frequency words, inhibits the conditional probability of high frequency words, avoids text degradation, and improves the quality and security of stegotext. With the breakthrough of Large Language Model (LLM), researchers have applied it to the field of text steganography. LLM has a powerful text generation ability through the training of large-scale text data, which can generate texts with higher readability, semantics and security. Therefore, it gave birth to the emergence of text steganography based on LLM. Wang et al. [35] used LLM to generate stegotext for the first time. By using LLaMA-2 as the generative model, higher quality texts are generated. Qi et al. [36] used LLaMA-2 and baichuan-2 as language models for English and Chinese, respectively. By grouping the words with the same prefix relationship in the candidate pool, the random number generator is used to ensure that the sender and the receiver select the same word, which solves the problem of segmentation ambiguity and generates high security text. The comparison of their advantage and disadvantage is shown in Table 1.

Table 1. Comparison of distinct types of generative text steganography.

Type	Advantage	Disadvantage
Rule-based steganography [13–15]	It is easy to implement and fast to generate.	The content of the generated text is single, and the sentences are unrelated.
Markov-based steganography [17]	It can effectively capture the dependencies between adjacent words.	It has poor dependence on long sequences and is difficult to avoid specific steganalysis algorithms.
RNN-Stega [25]	It can effectively capture the longer dependencies between long-distance words.	It generates text that performs poorly in terms of statistical imperceptibility.
VAE-Stega [31]	It improves the performance of steganographic texts in terms of statistical imperceptibility.	It does not control the semantics of the generated text.
NLS [33]	1. It can effectively reduce the gap between the distribution of normal text and stegotext. 2. it can constrain the semantics of generated text.	It has increased runtime and model training costs.
SAAC [10]	The imperceptibility of the text it generates is further reduced	Its predictive accuracy declined slightly
LLsM [35]	1. It generates high quality text. 2. The introduction of cue words improves the language controllability of the generated text.	1. Inaccurate capture of the core content of the prompt text. 2. High training and deployment costs

All the above generative text steganography can embed secret information in the stegotext generation process. However, it too pursues the quality of part of the text, ignoring the situation that the generated words deviate from the context semantics in the process of text generation. With the increase of the length of the secret information, the length of the generated text increases, which leads to the generation of semantically confusing text, and the verification affects the security of the stegotext.

3. Background of Text Steganography

3.1. The Framework for Text Steganography

The general framework of text steganography can be described vividly by the Simmons' "prisoner's problem" [37], as shown in Figure 1. Alice, as the sender, needs to hide the secret information and send it to the receiver Bob. Then Bob decrypts and restores the received secret information. At the same time of the transmission process, there is a "monitor" Eve, who is responsible for monitoring the transmitted text and judging whether it contains secret information.

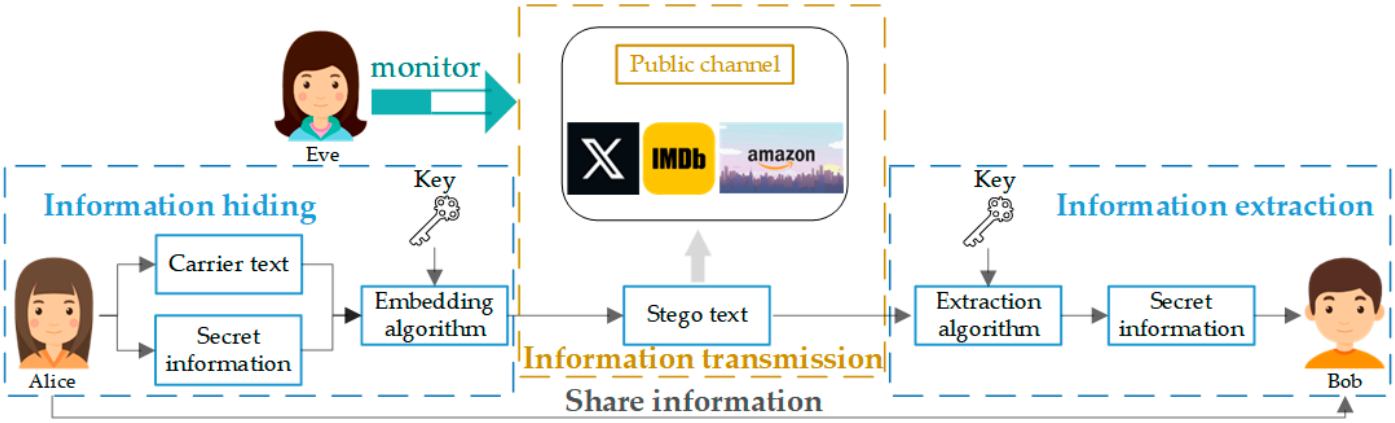


Figure 1. The general framework of steganography.

The model was modeled with the equation shown in Equation (1) [25]:

$$\begin{aligned} Emb &: C \times K \times M \rightarrow S \\ Ext &: S \times K \rightarrow M \end{aligned} \tag{1}$$

where *Emb* is information hiding, *Ext* is information extraction, C is the carrier text to hide the secret information, K is the key to encrypt or contain the shared information, M is the secret information to be hidden, S is the stegotext.

3.2. Generative Text Steganography

Generative text steganography is a technique that uses NLP to generate text containing secret information. In NLP, the process of text generation is usually viewed as the creation of a sequence $x = [x_1, \ldots, x_n]$ based on the joint probability of the language model $P_{LM}(x)$, which is formulated as follows [38]:

$$P_{LM}(x) = \prod_{t=1}^{n} P_{LM}(x_t | x_1, \ldots, x_{t-1}) = \prod_{t=1}^{n} P_{LM}(x_t | x_{<t}) \tag{2}$$

Different from the traditional text steganography, generative text steganography does not need the original cover, but directly embedded the secret information in the process of automatic text generation, which greatly improved the efficiency of secret information embedding [39]. With the development of neural network models, generative

text steganography based on neural network models has become a hot topic in the current field of research. The generation process is shown in Figure 2.

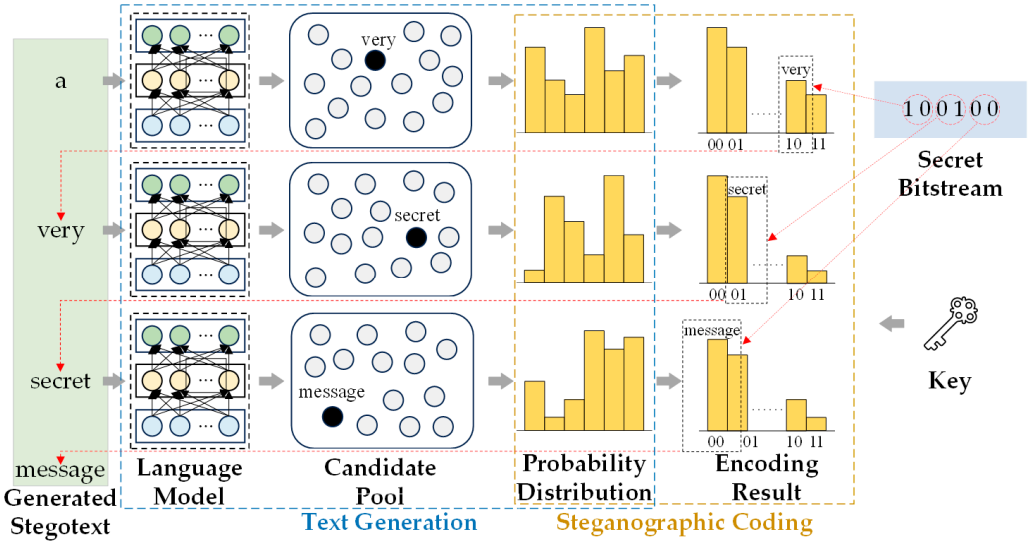


Figure 2. Generative process of generative text steganography based on neural network model.

In the stegotext generation stage, the neural network model used as text generation is first trained. Then, according to the current text sequence, the trained neural network model is used to predict the word, and the candidate word pool is constructed. Finally, under the constraints of secret information bits, encoding strategy and key, the candidate words that meet the requirements are selected. The above process is repeated until the secret information is completely embedded, and the stegotext is obtained [40].

With the emergence of large-scale pre-trained language models based on transformers, especially the wide application of LLM, the quality of generated text is getting higher and higher, and generation-based text steganography is more and more applied in the field of text steganography [39].

3.3. Steganographic Coding Strategy

For generative text steganography, a steganographic coding strategy is necessary to ensure that the receiver can extract secret information from the stegotext with less shared information. Taking FLC and HC as examples [25], the specific structure is shown in Figures 3 and 4.

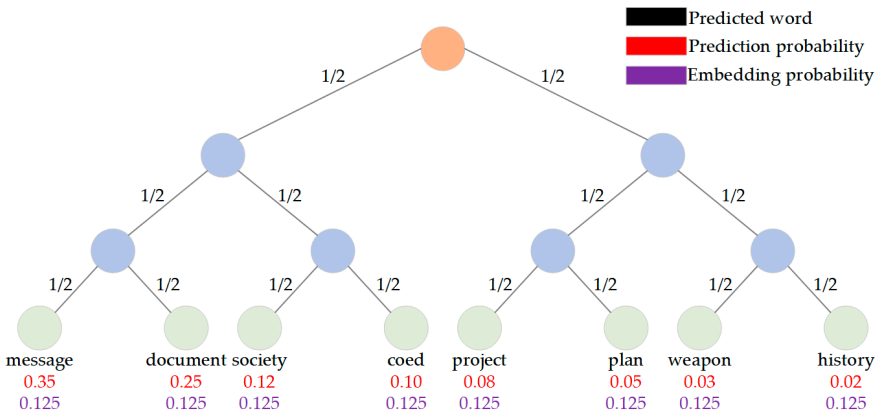


Figure 3. Structure of fixed-length coding.

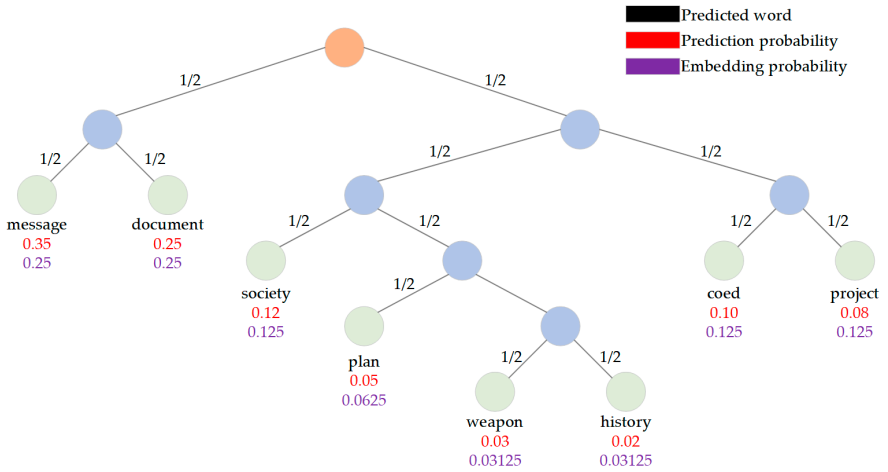


Figure 4. Structure of Huffman coding.

In the secret information embedding phase, the binary tree of the corresponding encoding is constructed, and the candidate words corresponding to the encoded values are selected. On the premise that the sender and the receiver share the text generation model and steganographic coding strategy, the receiver encodes the candidate pool in each step of the text generation process, and according to the stegotext, the secret information bit stream corresponding to the current word is obtained. In the stegotext generation process, the words with high prediction probability should be selected as much as possible. Therefore, the choice of encoding strategy has a great impact on the quality of stegotext [41].

4. Emotionally Controllable Text Steganography Framework Based on Large Language Model and Named Entity

In this paper, we propose an emotionally controllable text steganography based on LLM and named entity recognition, and the overall architecture is shown in Figure 5. The core of the framework can be split into three parts, which are the entity extraction module for obtaining named entities, the sentiment analysis module for obtaining the sentiment tendency of the current context, and the steganography generator for generating stegotext.

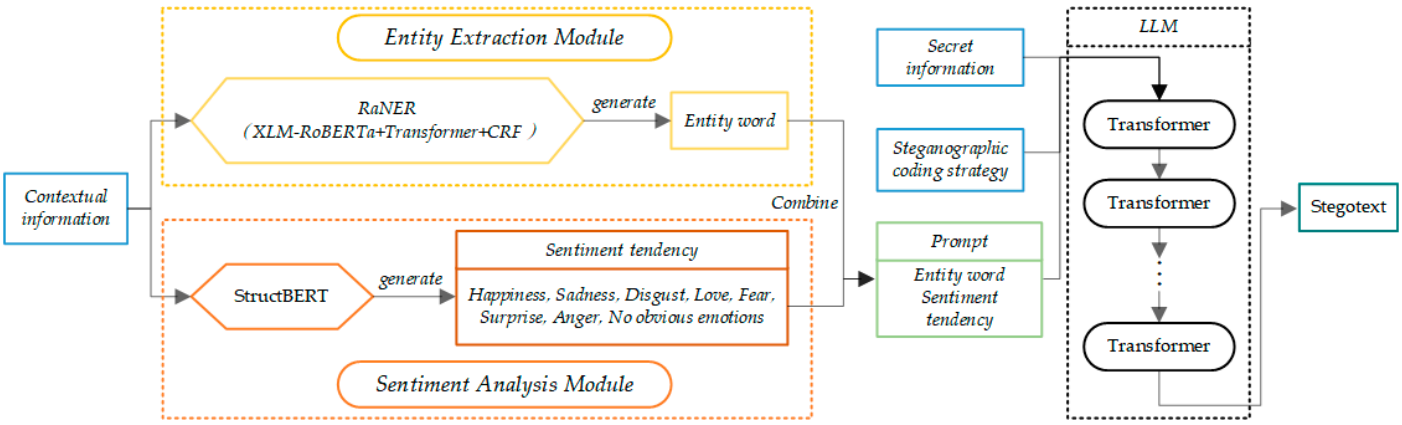


Figure 5. The overall architecture of the proposed method.

4.1. Entity Extraction Module

In order to better convey the secret information to the receiver and reduce the possibility of the generated stegotext being perceived by the "monitor" Eve, it is necessary to improve the semantic correlation between the current context and the stegotext. Therefore, before the LLM generates text based on secret information, it is provided with a set of cue

words. Through the entity extraction method, the entities are extracted from the current context information, and the entity words with different numbers are randomly selected to fill in the cue words, and the encoding results are used as the input of the LLM.

In this paper, the retrieval-augmented named entity recognition (RaNER) model [42] is chosen as the entity extraction module. The specific structure is shown in Figure 6.

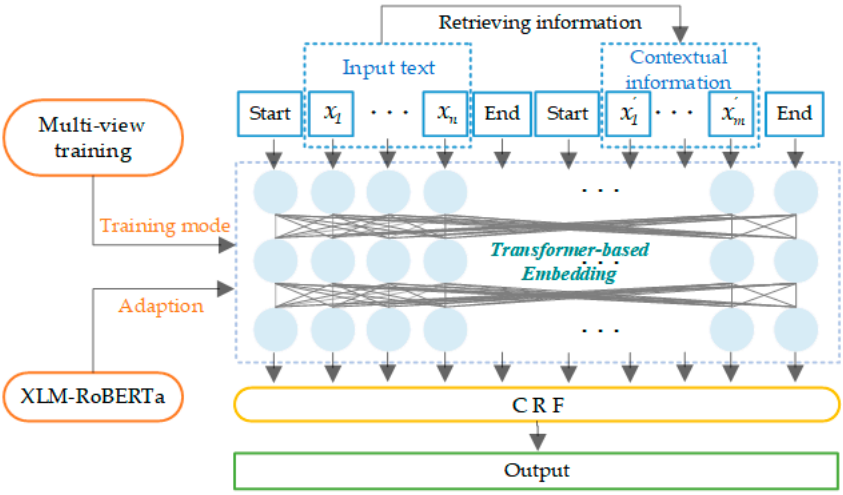


Figure 6. Structure of entity extraction module.

RaNER uses Transformer and conditional random field (CRF) as the base model and combines cross-lingual language model (XLM) [43] with robustly optimized BERT pretraining approach (RoBERTa) [44] is combined as a pretraining model base, by following the Transformer architecture and dynamic masking strategy of RoBERTa, while inheriting the XLM pair sentence structure, the model training efficiency becomes faster, the text generation is adapted to more linguistic scenarios, and the entity recognition is expanded from a monolingual BERT relies on local context to combine with global retrieval of relevant knowledge. External tools are used to retrieve external sentences related to the current context as additional context, which provides supplementary information to solve the problem of polysemy and improves the accuracy of entity recognition in the whole text. Finally, a multi-view training approach is adopted to improve the performance of named entity recognition by integrating the representations of the original text and the semantic information of external sentences retrieved by external tools. According to the generic domain named entities, four entity categories are preset, as shown in Table 2 [24], where MISC are all other entities [45].

Table 2. The predefined entity categories in RaNER.

Entity Categories	Entity Code
Location	LOC
Organization	ORG
Person	PER
Miscellany	MISC

The entity extraction module is a process from entity extraction to entity selection and finally to prompt template generation, as shown in Table 3, an example of entity extraction from secret text to backfill prompt words. The process of entity word selection is random, depending on the number of extraction results. If the number of entity identification is 0, then there is no selection. If 1, then select the entity word. When the entity word is greater than or equal to 2, then randomly select half the number of entity words.

Table 3. Example of entity extraction module.

Entity Extraction Process	Content
Current contextual information	I cant compare this movie with anything else, maybe except the movie "Leon" wonderfully played by Jean Reno and Natalie Portman
Extraction result	{'output': [{'type': 'PER', 'start': 98, 'end': 107, 'prob': 0.98851204, 'span': 'Jean Reno'}, {'type': 'PER', 'start': 112, 'end': 128, 'prob': 0.9671089, 'span': 'Natalie Portman.'}]}
Selection of entities	'Jean Reno'/'Natalie Portman.'
Template filling	Entity word: <Person>Jean Reno</Person >

4.2. Sentiment Analysis Module

The inclusion of a sentiment analysis module in the text steganography framework is to ensure that the generated stegotext and the current context can maintain consistency in terms of sentiment tendency, so as to improve the cognitive imperceptibility of the generated results. The pre-trained model StructBERT [46] is selected as the sentiment analysis model in the framework. The specific model structure is shown in Figure 7.

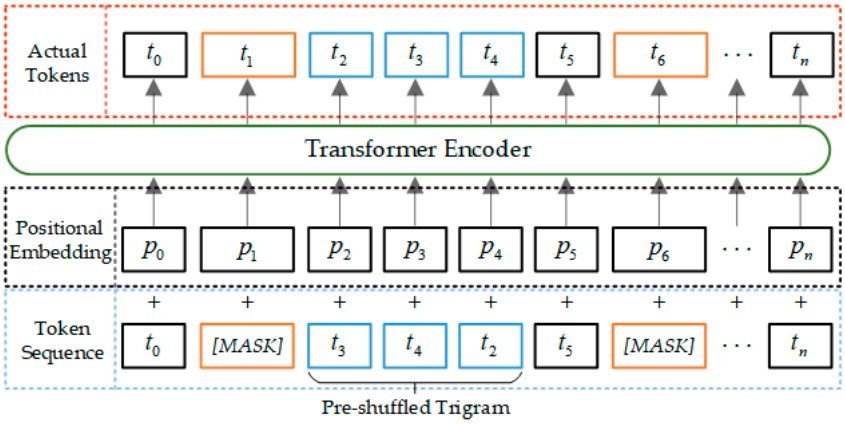


Figure 7. Structure of sentiment analysis module.

By disrupting some of the words in the sentence and requiring the model to restore the original order, the model is forced to understand the dependency relationship between words and the local structure, and to capture the sentiment cues in the complex sentences more accurately, so as to enhance the model's capability of capturing the sentiment expression of the utterance.

As shown in Table 4, by classifying the current context into happy, sad, disgusted, and other emotions [46], the module backfills the identified emotion categories into the cue word template to realize the emotion injection into the implicitly written text.

Table 4. Example of sentiment analysis module.

Sentiment Analysis Process	Content
Current Context Information	I cant compare this movie with anything else, maybe except the movie "Leon" wonderfully played by Jean Reno and Natalie Portman
Extraction result	{'scores': [0.015, 0.041, 0.046, 0.615, 0.221, 0.052, 0.009], 'labels': ['surprise', 'happiness', 'sadness', 'love', 'disgust', 'angry', 'fear']}
Template Filling	Sentiment tendency: <emotion>preferred</emotion>

4.3. Stego Generator

The steganography generator consists of the steganography encoding module and the text generation module. The steganographic coding module generates the corresponding coded values based on the conditional probability of the candidate word at each moment under a specific coding strategy. HC has been widely used in text steganography due to its adoption of variable-length encoding and its property of having a better compression ratio [25]. From [9,10], HC has an advantage in perceptual imperceptibility compared with coding methods such as AC and SAAC. The powerful text generation capability of LLM can effectively make up for the shortcomings in statistical and semantic imperceptibility. Therefore, in this paper, HC is chosen as the coding strategy for generating stegotext, and its implementation is shown as follows [47]: by constructing a special binary tree (called Huffman tree), the high frequency characters are mapped to shorter binary codes, and the low frequency characters are mapped to more extended binary codes by taking advantage of the difference in the frequency of characters' appearances in order to achieve the purpose of compressing data. Each leaf node corresponds to a character in the tree, and non-leaf nodes represent the merging of character codes. Character encodings with high frequency are located on shorter paths, while character encodings with low frequency are located on longer paths, thus increasing the probability of selecting high frequency characters.

The text generation module uses an LLM as a language model for the auto-regressive generation of text sequences. Firstly, the prompts of the LLM are obtained based on the contextual information, entity extraction module and sentiment analysis module. The prompt template is shown in Table 5. The LLM generates a candidate pool at each moment under the constraint of the prompts. Subsequently, the steganographic encoding module generates an encoding value for each candidate word, and determines the candidate word based on the mapping relationship between secret information and coded value. Until the secret information is fully embedded, the stegotext is generated. The proposed information hiding algorithm are shown in Algorithm 1.

Algorithm 1 Information hiding algorithm

Input: Secret information B, Contextual information I_{ctx}
Output: Stegotext T
Initialize T
while B is not empty **do**
  $P_T \leftarrow$ Generate text prompts based on I_{ctx}
  $w_E \leftarrow$ Entity words are obtained from I_{ctx} through the entity extraction
  If $len(w_E) = 1$ **then**
    $P_E \leftarrow$ Generates entity prompt based on w_E
  else
    $P_E \leftarrow$ Generates entity prompts based on $\lceil len(w_E) \rceil$ of w_E
  end if
  $w_S \leftarrow$ Sentiment words are obtained from I_{ctx} through the sentiment analysis
  $P_S \leftarrow$ Generates entity prompt based on w_S
  Prompts for LLM input $P = P_T + P_E + P_S$
  $c_{pool} \leftarrow$ Generate the candidate pool at current step t based on the P and T
  $HT \leftarrow$ Build a Huffman tree based on c_{pool}
  $b \leftarrow$ Obtain the bitstream of the embedding at t based on B and HT
  $x_t \leftarrow$ Generate the output word based on c_{pool}, HT, b
  $B = B - b$, $T = T + x_t$, $t = t + 1$
end while
return Stegotext T

Table 5. The prompt template of LLM.

Keys	Content
Description	Template used by our model
Prompt input	Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.\n\n Instruction:\n{instruction}\n\n Input:\n{input}\n\n Response:\n
Prompt no input	Below is an instruction that describes a task. Write a response that appropriately completes the request.\n\n Instruction:\n{instruction}\n\n Response:\n
Response split	Response:
Instruction	You specialise in imitating human discourse in four areas: vocabulary, intonation, syntax and semantics. The user will enter text. You will need to create a sentence similar to it. The generated text should be more focused on 'Entity words'. The sentiment of the generated text is 'Sentiment tendency'.
Input	Contextual information, Entity words, Sentiment tendency
Response	Output similar to the input text

5. Experiments and Analysis

5.1. Dataset and Experimental Setup

To be able to extract the central text and sentiment features from the text, this paper selects the IMDB movie review dataset [48] as the evaluation dataset. The IMDB movie review dataset is a widely used data resource for sentiment analysis and natural language processing tasks. After a series of preprocessing operations such as text cleaning and removing stop words, 33,000 pieces of data are obtained. The data set was divided 9:1:1, and 27,000 pieces of data were used as the training set for training, 3000 pieces of data were used as the test set to test the trained model, and the remaining 3000 pieces of data were used as the validation set. As shown in Figure 8, the median length of the 27,000 IMDB movie review dataset is 106, the average length is 131.7, and over 95% of the texts are less than 500 characters in length.

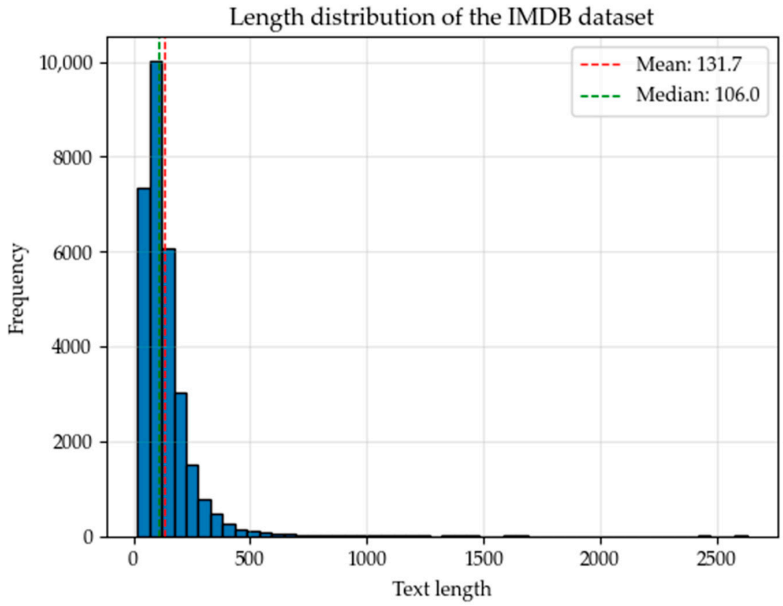


Figure 8. Length distribution of the IMDB dataset.

The experiments in this paper are conducted based on Pytorch 2.4.0 and Python 3.10, and the model training is accelerated by using NVIDIA A10 and CUDA 12.4. To ensure a fair comparison, we rebuild all baseline methods using the same details throughout the entire experiment. For RNN-Stega [25] and VAE-Stega [31] to be trained, the word vector dimension is 768, the hidden state dimension is 768, the dropout is set to 0.5, and the number of model training epochs is 50. The batch size is 128, the learning rate is 0.0001, and the maximum character length of the generated text does not exceed 256. Pretrained models (e.g., GPT-2, Qwen-2.5 and LLaMA-3) are obtained using Huggingface library.

5.2. Evaluation Metrics

5.2.1. Hidden Capacity

Hidden capacity is an important metric of text steganography. It describes the amount of secret information embedded in the stegotext. Hidden capacity is usually evaluated by Embedding Rate (ER) [34], which is calculated as shown in Equation (3):

$$ER = \frac{1}{N}\sum_{i=1}^{N}\frac{L_i}{S_i} \tag{3}$$

where N represents the number of generated sentences, L_i is the number of bits embedded in the ith sentence, and S_i is the length of the ith sentence.

5.2.2. Perceptual Imperceptibility

In order to ensure the perceptual imperceptibility of the stegotext, the stegotext is required to be perceptually similar to the real text. The perceptual imperceptibility evaluation of text steganography can be carried out from both subjective and objective perspectives. Subjective evaluation means that people rate the stegotext based on its fluency and semantic coherence. The objective evaluation relies on the standard measure of text quality evaluation, perplexity (PPL) [25], to quantify the imperceptibility of the stegotext. The smaller the value, the higher the quality of the generated text. The specific calculation formula is shown in Equation (4):

$$ppl = 2^{-\frac{1}{N}\sum_{i=1}^{N} logP_i(S_i)} \tag{4}$$

where N represents the total number of stegotext sentences, $S_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ is the ith sentence of the generated stegotext, n represents the total number of words in the ith sentence, and $P_i(S_i)$ represents the probability of the ith sentence.

5.2.3. Security

The difference in statistical distribution between the stegotext and the real text directly characterizes the security of the steganography algorithm. Kullback-Leibler divergence (KLD) [11] is usually used to calculate the difference between two probability distributions in terms of information entropy and other features, and the specific calculation formula is shown in Equation (5):

$$D_{KL}(P\|Q) = \sum_{i=1}^{N}[p(x_i)logp(x_i) - p(x_i)logq(x_i)] \tag{5}$$

where P and Q represent the distribution before and after encoding, $p(x_i)$ and $q(x_i)$ represent the ith probability value in the distributions P and Q, respectively, and N represents the total number of stegotexts.

5.2.4. Cognitive Imperceptibility

When evaluating cognitive imperceptibility, it is necessary to determine whether the content and emotional features of the stegotext are related to the current context content. Therefore, this paper adopts bilingual evaluation understudy (BLEU) [49] to evaluate cognitive imperceptibility. It evaluates the quality of the generation by calculating the degree of n-gram overlap between the generated text and one or more reference texts. A higher BLEU value indicates a higher quality of the generated text. The calculation formula is shown in Equation (6) [50]:

$$BLEU = BP \times exp(\sum_{n=1}^{N} w_n log(P_n)) \tag{6}$$

where BP is the penalty factor, which is used to prevent the model from falsely improving the accuracy by repeatedly generating part of the content. w_n is the weight used to weight the precision of different n-grams when calculating the BLEU score. P_n is the coincidence accuracy between the generated stegotext and the reference text. The specific calculation formulas of BP and P_n are shown in Equations (7) and (8):

$$BP = \begin{cases} 1, l_c > l_s, \\ e^{1-\frac{l_s}{l_c}}, l_c \leq l_s \end{cases} \tag{7}$$

$$P_n = \frac{\sum_i \sum_k min(h_k(c_i), max_{j \in m} h_k(s_{ij}))}{\sum_i \sum_k min(h_k(c_i))} \tag{8}$$

where l_c is the length of the generated stegotext, l_s is the length of the reference text, $h_k(c_i)$ is the number of occurrences of the kth n-gram in the generated stegotext c_i of reference text i, $h_k(s_{ij})$ is the number of occurrences of the kth n-gram in the corresponding reference text s_j of reference text i, and m is the set of reference texts.

5.3. Comparison Experiment

In this paper, experiments are carried out on the IMDB movie review dataset [48], and the representative LLaMA-3 and Qwen-2.5 in the large model are selected as text generation models for empirical research on text steganography. In the comparison experiments, two types of classical methods are selected respectively. One is represented by RNN-Stega [25] and VAE-Stega [31], which need to be trained deep learning model methods. NLS [33], SAAC [10], and LLsM [35] are pretrained methods that require no additional training.

In the comparison experiments, the dataset is divided into training, testing, and validation sets by 9:1:1. The models to be trained are trained for 50 rounds, and then experiments are conducted separately for all models using the validation set. In the stegotext generation process, stegotext is generated by randomly generating a bit stream of 30 to 256 bits as secret information. For the affective controlled text steganography approach fusing LLM and named entity proposed in this paper, based on hiding the above secret information, the validation set is used as the information of the current context for prompt word generation, which is tested on models of different parameter sizes, such as Qwen-2.5, LLaMA-3, and the specific results are shown in Table 6.

It can be seen from the comparison experiments that the LLaMA-3 (3B), which integrates the entity extraction and sentiment analysis modules, achieves the optimum in three evaluation metrics, ER, PPL, and BLEU, in the results of the text steganography experiments conducted. The optimal model for the other evaluation metrics, KLD, is Qwen-2.5 (3B), which integrates the entity extraction and sentiment analysis modules. The model with unconstrained semantics constructs the candidate pool by considering only the

influence brought by the words around the predicted position, resulting in a deviation of the probability distribution of the candidate words from the probability distribution of the normal text at that position. After the introduction of the entity extraction module and the sentiment analysis module, the construction of the candidate pool at the predicted position is more comprehensively considered, and the probability distribution is closer to that of the normal text in terms of probability distribution, which makes the KLD smaller and improves the security of the text.

Table 6. The results of comparing experimental.

Methods	Evaluation Metrics			
	ER (bit/word)	**PPL**	**KLD**	**BLEU**
RNN-Stega [25]	1.374	28.763	28.406	18.634
VAE-Stega [31]	1.156	34.598	23.675	15.764
NLS [33]	1.592	21.286	20.938	23.079
SAAC [10]	1.731	26.474	17.373	17.323
LLsM [35]	2.138	17.865	18.984	65.781
Entity Extraction + Sentiment Analysis + Qwen-2.5 1.5B	1.877	19.338	18.672	44.336
Entity Extraction + Sentiment Analysis + Qwen-2.5 3B	2.256	18.092	**17.336**	62.134
Entity Extraction + Sentiment Analysis + LLaMA-3 1B	1.950	16.872	17.351	59.697
Entity Extraction + Sentiment Analysis + LLaMA-3 3B	**2.383**	**16.304**	17.952	**68.864**

Bold values indicate the best performance.

Taken together, the LLM integrating the entity extraction and sentiment analysis modules achieves results ahead of the comparative models in the text steganography methods, and it can be proved that the model has better results in embedding ability, hiddenness, and security.

5.4. Anti-Steganalysis

To assess the security of steganography, a steganalyzer is usually used to distinguish between normal text and stegotext. we use three steganalysis models: LS-CNN [23], TS-RNN [51] and BERT classifier [52] to test the anti-steganalysis ability of each model. The evaluation metrics are accuracy (Acc) and recall (R). The calculation formula is as follows:

$$\text{Acc} = \frac{T_p + T_n}{F_p + F_n + T_p + T_n} \tag{9}$$

$$\text{R} = \frac{T_p}{T_p + F_n} \tag{10}$$

where T_p is the number of stegotexts correctly judged by steganalysis as stegotexts, T_n is the number of normal texts correctly judged by steganalysis as normal texts, F_p is the number of stegotexts incorrectly determined by steganalysis as normal texts, and F_n is the number of normal texts incorrectly determined by steganalysis as stegotexts.

Table 7 shows the anti-steganalysis comparison of the proposed model and the baseline model. According to Table 5, the proposed model shows a good performance in anti-steganalysis, which improves the security of stegotext.

5.5. Ablation Experiment

In order to further validate the effectiveness of the entity extraction module and sentiment analysis module in the text steganography framework proposed in this paper,

this paper conducts ablation experiments with LLaMA-3 as the language model, and the specific experimental results are shown in Table 8.

Table 7. The results of anti-steganalysis experimental.

Methods	LS-CNN [23]		TS-RNN [51]		BERT [52]	
	ACC	R	ACC	R	ACC	R
RNN-Stega [25]	0.958	0.957	0.921	0.930	0.936	0.944
VAE-Stega [31]	0.917	0.922	0.893	0.908	0.933	0.931
NLS [33]	0.851	0.844	0.853	0.857	0.877	0.853
SAAC [10]	0.862	0.853	0.881	0.866	0.896	0.900
LLsM [35]	0.732	0.740	**0.704**	0.70.7	0.764	0.758
Entity Extraction + Sentiment Analysis + Qwen-2.5 1.5B	0.833	0.875	0.819	0.801	0.831	0.844
Entity Extraction + Sentiment Analysis + Qwen-2.5 3B	0.737	0.758	0.742	0.745	0.741	0.739
Entity Extraction + Sentiment Analysis + LLaMA-3 1B	0.805	0.831	0.803	0.821	0.812	0.801
Entity Extraction + Sentiment Analysis + LLaMA-3 3B	**0.717**	**0.730**	0.709	**0.698**	**0.738**	**0.739**

Bold values indicate the best performance.

Table 8. The results of ablation experimental.

Group	Module		Evaluation Metrics			
	Entity Extraction	Sentiment Analysis	ER (bit/word)	PPL	KLD	BLEU
1	×	×	1.905	19.127	19.210	40.691
2	√	×	**2.921**	17.847	19.941	54.369
3	×	√	2.016	18.581	18.504	50.108
4	√	√	2.383	**16.304**	**17.952**	**68.864**

Bold values indicate the best performance.

The comparison of the results of each model under different module combinations is shown in Figure 9. The horizontal axis represents the different combination methods, the left side of the vertical axis represents the scores of ER, KLD, and PPL, and the right side represents the BLEU scores. The results show that the LLaMA-3 (3B), which incorporates the entity extraction and sentiment analysis modules, dominates the evaluation metrics of PPL, KLD, and BLEU. In the experimental results, the introduction of entity extraction module and sentiment analysis module is mainly reflected in BLEU. Through entity words and sentiment tendency, the language model is guided to predict the candidate pool of the target location, and more consideration is given to the contextual information. The generated stegotext matches the context information better, which enhances the semantic concealment. The LLM without incorporating the entity extraction and sentiment analysis module consistently lagged the metrics compared to the other three methods. The above results show the effectiveness of the entity extraction module and the sentiment analysis module for generating controlled stegotext.

To verify the generalization of the module proposed in this paper, LLaMA-3 (1B) with smaller parameters is tested and the results are shown in Table 9. From the results, it is easy to see that the introduction of the module achieves better results in all four metrics, ER, PPL, KLD and BLEU, compared to pure LLaMA-3 (1B). This shows that the module designed in this paper has better generalization and proves that our method still has better results at lower scales.

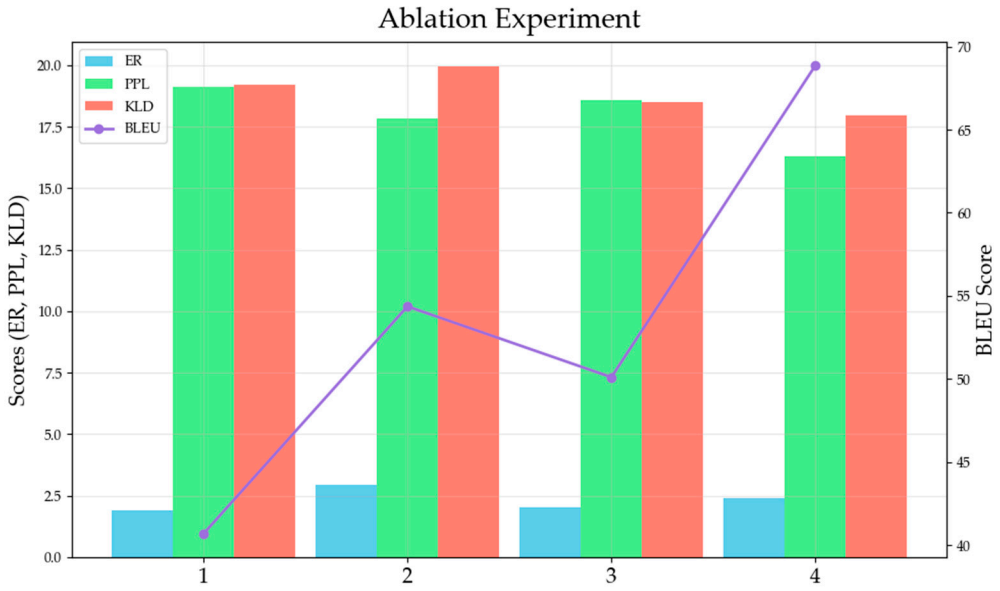


Figure 9. The results of ablation experimental.

Table 9. Experimental results on module generalization.

Group	Module		Evaluation Metrics			
	Entity Extraction	**Sentiment Analysis**	**ER (bit/word)**	**PPL**	**KLD**	**BLEU**
1	×	×	1.878	19.233	19.988	42.580
2	√	×	**2.209**	18.455	18.024	51.721
3	×	√	1.805	**16.787**	19.655	52.643
4	√	√	1.950	16.872	**17.351**	**59.697**

√ indicates that the corresponding module has been used. × indicates that the corresponding module has not been used. Bold values indicate the best performance.

5.6. Case Study

As shown in Table 10, we sample several stegotexts generated by baselines and our model. We can find that our stegotext performs better semantically and emotionally compared to the baseline. This indicates that our model has great potential for imperceptibility.

Table 10. Case Study on the IMDB datasets.

Methods	Generated StegoText
RNN-Stega [25]	You have never been a little too hard to understand why. The movie was so good that I hoped that the movie was better and better than that it was.
VAE-Stega [31]	You said Sadakooooooooooo is a mysterious character. It was a good film. it was the first time I had seen the movie in my childhood. This movie is worth your eyes.
NLS [33]	The film itself is a bit of a mystery. The story is mostly about Sadako's problems, but it does a good job of explaining why she's shy and what her problems are.
SAAC [10]	It is a good movie, which is good, courtesy of Yukie Namam. Not a single one of it created some adoration that most of us, myself included, could get good.

Table 10. *Cont.*

Methods	Generated StegoText
LLsM [35]	You might debate whether the enigma of Sadako should be left alone, but this film is quite compelling. The lead, played by Yukie Nakama as a reserved Sadako, and burdened with abilities she never chose.
Entity Extraction + Sentiment Analysis + LLaMA-3 3B	While some may debate if Sadako's enigma should stay unresolved, this film is undoubtedly excellent. Yukie Nakama shines as the shy lead, a compelling performance reminiscent of Carrie White but with her own captivating charm.

To assess the quality of the production of stegotexts, we randomly selected 10 sets of texts from the results and invited 10 PhDs to score them on a 5-point scale. The results are shown in Table 11. For the human evaluation, in general, the steganography based on LLM generation has achieved high scores. Compared with [35], the two modules proposed in this paper enable the generator to consider contextual information more comprehensively, and make the generated text maintain fluency while having better performance in semantics, so the score is slightly higher than it.

Table 11. The results of human evaluation.

Methods	Point
RNN-Stega [25]	2.32
VAE-Stega [31]	2.15
NLS [33]	3.56
SAAC [10]	3.09
LLsM [35]	4.17
Entity Extraction + Sentiment Analysis + LLaMA-3 3B	**4.33**

Bold values indicate the best performance.

6. Conclusions

In this paper, we focus on generative text steganography based on LLM and propose an emotionally controllable text steganography based on LLM and named entity. In this method, an entity extraction module composed of RaNER and a sentiment analysis module composed of StructBERT are designed to extract the central text and sentiment features from the current context information respectively. It is then used as the input of the large model, which ensures the controllable generation of stegotext. Compared with the existing mainstream algorithms under the public data set, the results show that the proposed method is effective in hiding capacity, concealment and security. Specifically, the LAMA-3 (3B) model which combines the entity extraction module and the sentiment analysis module has an embedding rate of 2.383 bit per word, a PPL of 16.304, a KLD of 17.952, and a BLEU of 68.864. Ablation experiments show that the entity extraction module and sentiment analysis module designed in this paper can effectively improve the hiding capacity, invisibility and security of stegotext.

Although the proposed method performs better overall, we recognize that there are still challenges with our method. The method currently achieves good performance in terms of semantics, but there are opportunities for improvement in terms of hiding capacity and security. Therefore, in the future research, one is to try to adopt the optimized steganographic coding strategy to improve the hiding capacity of the generated text. Secondly,

the relationship between the distribution of real text and the distribution of stegotext can be considered, and the secure transmission of stegotext can be achieved by reducing the statistical gap between them. In addition, LLM itself brings huge computational overhead and has higher requirements for computing hardware. The introduction of entity extraction module and sentiment analysis module further increases the computational overhead. Whether the model can be improved to run on smaller devices becomes another direction for future research.

Author Contributions: Conceptualization, H.S. and W.G.; methodology, H.S. and W.G.; software, H.S. and S.G.; validation, H.S. and S.G.; formal analysis, H.S. and S.G.; investigation, H.S.; resources, W.G.; data curation, H.S. and S.G.; writing—original draft preparation, H.S.; writing—review and editing, W.G.; supervision, W.G.; project administration, W.G.; funding acquisition, W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in a GitHub repository at https://github.com/SHZXX109/dataset_IMDB (accessed on 5 May 2025).

Acknowledgments: The authors gratefully acknowledge the reviewers for their precious time and effort dedicated to this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Keys	Content
AC	Arithmetic coding
ACC	Accuracy
BERT	Bidirectional encoder representations from transformers
BLEU	Bilingual evaluation understudy
CRF	Conditional random field
DL	Deep learning
DNN	Deep neural network
ER	Embedding rate
FLC	Fixed-length coding
GAN	Generative adversarial network
HC	Huffman coding
KLD	Kullback-Leibler divergence
LLM	Large language model
LSTM	Long short-term memory
NLP	Natural language processing
PPL	Perplexity
R	Recall
RaNER	Retrieval-augmented named entity recognition
RNN	Recurrent neural network
RoBERTa	Robustly optimized BERT pretraining approach
stegotext	Steganographic text
VAE	Variational auto-encoder
XLM	cross-lingual language model

References

1. Khan, I.; Ali, Q.E.; Hadi, H.J.; Ahmad, N.; Ali, G.; Cao, Y.; Alshara, M.A. Securing Blockchain-Based Supply Chain Management: Textual Data Encryption and Access Control. *Technologies* **2024**, *12*, 110. [CrossRef]
2. Xu, Y.; Liu, J.; You, Z.; Zhang, T. A Novel Color Image Encryption Algorithm Based on Hybrid Two-Dimensional Hyperchaos and Genetic Recombination. *Mathematics* **2024**, *12*, 3457. [CrossRef]
3. McAteer, I.; Ibrahim, A.; Zheng, G.; Yang, W.; Valli, C. Integration of Biometrics and Steganography: A Comprehensive Review. *Technologies* **2019**, *7*, 34. [CrossRef]
4. Rashid, A.; Salamat, N.; Prasath, V.B.S. An Algorithm for Data Hiding in Radiographic Images and ePHI/R Application. *Technologies* **2018**, *6*, 7. [CrossRef]
5. Selma, T.; Masud, M.M.; Bentaleb, A.; Harous, S. Inference Analysis of Video Quality of Experience in Relation with Face Emotion, Video Advertisement, and ITU-T P.1203. *Technologies* **2024**, *12*, 62. [CrossRef]
6. Qin, Z.; Sun, C.; He, T.; He, Y.; Abdullah, A.; Samian, N.; Roslan, N.A. ADLM—stega: A Universal Adaptive Token Selection Algorithm for Improving Steganographic Text Quality via Information Entropy. *arXiv* **2024**, arXiv:2410.20825.
7. Cao, Y.; Zhou, Z.; Chakraborty, C.; Wang, M.; Wu, Q.M.J.; Sun, X. Generative Steganography Based on Long Readable Text Generation. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 4584–4594. [CrossRef]
8. Kuznetsov, O.; Chernov, K.; Shaikhanova, A.; Iklassova, K.; Kozhakhmetova, D. DeepStego: Privacy-Preserving Natural Language Steganography Using Large Language Models and Advanced Neural Architectures. *Computers* **2025**, *14*, 165. [CrossRef]
9. Wang, S.; Li, F.; Yu, J.; Lai, H.; Wu, S.; Zhou, W. Enhancing Semantic Consistency in Linguistic Steganography via Denosing Auto-Encoder and Semantic-Constrained Huffman Coding. In Proceedings of the 12th National CCF Conference on Natural Language Processing and Chinese Computing (NLPCC), Foshan, China, 12–15 October 2023; pp. 799–812.
10. Shen, J.; Ji, H.; Han, J. Near-imperceptible Neural Linguistic Steganography via Self-Adjusting Arithmetic Coding. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Virtual, 16–20 November 2020; pp. 303–313.
11. Ding, C.; Fu, Z.; Yang, Z.; Yu, Q.; Li, D.; Huang, Y. Context-Aware Linguistic Steganography Model Based on Neural Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 868–878. [CrossRef]
12. Badawy, I.L.; Nagaty, K.; Hamdy, A. A Comprehensive Review on Deep Learning-Based Generative Linguistic Steganography. In Proceedings of the 25th International Conference on Interactive Collaborative Learning (ICL), Vienna, Austria, 27–30 September 2022; pp. 651–660.
13. Meral, H.M.; Sankur, B.; Sumru, Ö.A.; Güngör, T.; Sevinç, E. Natural language watermarking via morphosyntactic alterations. *Comput. Speech Lang.* **2009**, *23*, 107–125. [CrossRef]
14. Xiang, L.; Wang, X.; Yang, C.; Liu, P. A novel linguistic steganography based on synonym run-length encoding. *IEICE Trans. Inf. Syst.* **2017**, *E100D*, 313–322. [CrossRef]
15. Lee, I.-S.; Tsai, W.-H. A new approach to covert communication via PDF files. *Signal Process.* **2010**, *90*, 557–565. [CrossRef]
16. Xiang, L.; Guo, G.; Yu, J.; Sheng, V.S.; Yang, P. A convolutional neural network-based linguistic steganalysis for synonym substitution steganography. *Math. Biosci. Eng.* **2020**, *17*, 1041–1058. [CrossRef] [PubMed]
17. Chen, X.; Sun, H.; Tobe, Y.; Zhou, Z.; Sun, X. Coverless information hiding method based on the Chinese mathematical expression. In Proceedings of the 1st International Conference on Cloud Computing and Security (ICCCS), Nanjing, China, 13–15 August 2015; pp. 133–143.
18. Wang, Y.; Song, R.; Li, L.; Zhang, R.; Liu, J. Dynamically allocated interval-based generative linguistic steganography with roulette wheel. *Appl. Soft Comput.* **2025**, *176*, 113101. [CrossRef]
19. Fang, T.; Jaggi, M.; Argyraki, K. Generating steganographic text with LSTMs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, Canada, 30 July–4 August 2017; pp. 100–106.
20. Gatt, A.; Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.* **2018**, *61*, 1–64. [CrossRef]
21. Chapman, M.; Davida, G. Hiding the hidden: A software system for concealing ciphertext as innocuous text. In Proceedings of the 1997 International Conference on Information and Communications Security (ICICS), Beijing, China, 11–14 November 1997; pp. 335–345.
22. Dai, W.; Yu, Y.; Dai, Y.; Deng, B. Text steganography system using Markov chain source model and DES algorithm. *J. Softw.* **2010**, *5*, 785–792. [CrossRef]
23. Wen, J.; Zhou, X.; Zhong, P.; Xue, Y. Convolutional Neural Network Based Text Steganalysis. *IEEE Signal Process. Lett.* **2019**, *26*, 460–464. [CrossRef]
24. Chen, Y.; Li, Q.; Wu, X.; Li, H.; Chang, Q. A Character-based Diffusion Embedding Algorithm for Enhancing the Generation Quality of Generative Linguistic Steganographic Texts. *arXiv* **2025**, arXiv:2505.00977.
25. Yang, Z.; Guo, X.; Chen, Z.; Huang, Y.; Zhang, Y. RNN-Stega: Linguistic Steganography Based on Recurrent Neural Networks. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1280–1295. [CrossRef]

26. Xiang, L.; Yang, S.; Liu, Y.; Li, Q.; Zhu, C. Novel Linguistic Steganography Based on Character-Level Text Generation. *Mathematics* **2020**, *8*, 1558. [CrossRef]
27. Yang, Z.; Wei, N.; Liu, Q.; Huang, Y.; Zhang, Y. GAN-TStega: Text Steganography Based on Generative Adversarial Networks. In Proceedings of the 18th International Workshop on Digital Forensics and Watermarking (IWDW), Chengdu, China, 2–4 November 2019; pp. 18–31.
28. Zhou, X.; Peng, W.; Yang, B.; Wen, J.; Xue, Y.; Zhong, P. Linguistic Steganography Based on Adaptive Probability Distribution. *IEEE Trans. Dependable Secure Comput.* **2022**, *19*, 2982–2997. [CrossRef]
29. Yang, Z.; Zhang, P.; Jiang, M.; Huang, Y.; Zhang, Y. RITS: Real-time interactive text steganography based on automatic dialogue model. In Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS), Haikou, China, 8–10 June 2018; pp. 253–264.
30. Yang, Z.; Xu, Z.; Zhang, R.; Huang, Y. T-GRU: ConTextual Gated Recurrent Unit model for high quality Linguistic Steganography. In Proceedings of the 2022 IEEE International Workshop on Information Forensics and Security (WIFS), Shanghai, China, 12–16 December 2022; pp. 1–6.
31. Yang, Z.; Zhang, S.; Hu, Y.; Hu, Z.; Huang, Y. VAE-Stega: Linguistic Steganography Based on Variational Auto-Encoder. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 880–895. [CrossRef]
32. Tu, H.; Yang, Z.; Yang, J.; Zhou, L.; Huang, Y. FET-LM: Flow-Enhanced Variational Autoencoder for Topic-Guided Language Modeling. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11180–11193. [CrossRef]
33. Ziegler, Z.M.; Deng, Y.; Rush, A.M. Neural linguistic steganography. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hongkong, China, 2–7 November 2019; pp. 1210–1215.
34. Pang, K.; Bai, M.; Yang, J.; Wang, H.; Jiang, M.; Huang, Y. FREmax: A Simple Method Towards Truly Secure Generative Linguistic Steganography. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 4755–4759.
35. Wang, Y.; Song, R.; Zhang, R.; Liu, J.; Li, L. LLsM: Generative Linguistic Steganography with Large Language Model. *arXiv* **2024**, arXiv:2401.15656.
36. Qi, Y.; Chen, K.; Zeng, K.; Zhang, W.; Yu, N. Provably Secure Disambiguating Neural Linguistic Steganography. *arXiv* **2024**, arXiv:2403.17524. [CrossRef]
37. Simmons, G.J. The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology; Proceedings of Crypto 83*; Springer: Boston, MA, USA, 1984; pp. 51–67.
38. Huang, Y.; Tian, C.; Narayanan, K.; Zheng, L. Relatively-Secure LLM-Based Steganography via Constrained Markov Decision Processes. *arXiv* **2025**, arXiv:2502.01827.
39. Xiang, L.; Wang, R.; Yang, Z.; Liu, Y. Generative Linguistic Steganography: A Comprehensive Review. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 986–1005. [CrossRef]
40. Pang, K. FreStega: A Plug-and-Play Method for Boosting Imperceptibility and Capacity in Generative Linguistic Steganography for Real-World Scenarios. *arXiv* **2024**, arXiv:2412.19652.
41. Wang, Y.; Pei, G.; Chen, K.; Ding, J.; Pan, C.; Pang, W.; Hu, D.; Zhang, W. SparSamp: Efficient Provably Secure Steganography Based on Sparse Sampling. *arXiv* **2025**, arXiv:2503.19499.
42. Wang, X.; Shen, Y.; Cai, J.; Wang, T.; Wang, X.; Xie, P.; Huang, F.; Lu, W.; Zhuang, Y.; Tu, K.; et al. DAMO-NLP at SemEval-2022 Task 11: A Knowledge-based System for Multilingual Named Entity Recognition. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval), Seattle, WA, USA, 14–15 July 2022; pp. 1457–1468.
43. Conneau, A.; Lample, G. Cross-lingual language model pretraining. In Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 8–14.
44. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
45. Wang, X.; Jiang, Y.; Bach, N.; Wang, T.; Huang, Z.; Huang, F.; Tu, K. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), Virtual, 1–8 August 2021; pp. 1800–1812.
46. Wang, W.; Bi, B.; Yan, M.; Wu, C.; Xia, J.; Bao, Z.; Peng, L.; Si, L. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–10.
47. Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proc. IRE* **1952**, *40*, 1098–1101. [CrossRef]
48. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), Portland, OR, USA, 19–24 June 2011; pp. 142–150.

49. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
50. Ding, C.; Fu, Z.; Yu, Q.; Wang, F.; Chen, X. Joint Linguistic Steganography With BERT Masked Language Model and Graph Attention Network. *IEEE Trans. Cogn. Dev. Syst.* **2024**, *16*, 772–781. [CrossRef]
51. Yang, Z.; Wang, K.; Li, J.; Huang, Y.; Zhang, Y.-J. TS-RNN: Text Steganalysis Based on Recurrent Neural Networks. *IEEE Signal Process. Lett.* **2019**, *26*, 1743–1747. [CrossRef]
52. Peng, W.; Zhang, J.; Xue, Y.; Yang, Z. Real-Time Text Steganalysis Based on Multi-Stage Transfer Learning. *IEEE Signal Process. Lett.* **2021**, *28*, 1510–1514. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.