Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010450
Article

Tech Science Press

Coverless Text Hiding Method Based on Improved Evaluation Index and One-Bit Embedding

Ning Wu[1,2], **Yi Yang**[1,*], **Lian Li**[1], **Zhongliang Yang**[3], **Poli Shang**[4], **Weibo Ma**[5] **and Zhenru Liu**[5]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou, 730000, China
[2]School of Computer Science & Artificial Intelligence Lanzhou Institute of Technology, Lanzhou, 730050, China
[3]Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
[4]School of Electronic and Electrical Engineering, Lanzhou Petrochemical Polytechnic, Lanzhou, 730060, China
[5]School of Electrical Engineering, Lanzhou Institute of Technology, Lanzhou, 730050, China
[*]Corresponding Author: Yi Yang. Email: yy@lzu.edu.cn

Received: 05 March 2020; Accepted: 04 June 2020

Abstract: In the field of information hiding, text is less redundant, which leads to less space to hide information and challenging work for researchers. Based on the Markov chain model, this paper proposes an improved evaluation index and one-bit embedding coverless text steganography method. In the steganography process, this method did not simply take the transition probability as the optimization basis of the steganography model, but combined it with the sentence length in the corresponding nodes in the model to gauge sentence quality. Based on this, only two optimal conjunctions of the current words are retained in the method to generate sentences of higher quality. Because the size of the training text dataset is generally large, this leads to higher complexity of the steganographic model; hence, fewer repetitions of the generated steganographic sentences occur. Different datasets and methods were selected to test the quality of the model. The results indicate that our method can achieve higher hiding capacity and has better concealment capability.

Keywords: Coverless text steganography; Markov chain; evaluation index; one-bit embedding

1 Introduction

Important and confidential information is constantly transmitted through public communication channels, which traditionally require the use of encryption technology. However, encryption technology has a major disadvantage when using this changed and incomprehensible information, even ordinary people can guess that there is a secret, let alone an enemy. Accordingly, the disclosure of secrets may only be a matter of time. There is a more subtle way to protect the transmission of secrets. For example, poetry has a long history in China, and it involves various forms with different rhythm rules. In ancient China, people often used this type of carrier to hide and transmit their secrets in different places in a poem that looked very beautiful to others. For example, an acrostic is a poem in which the first word of each sentence is stitched together sequentially to form a complete expression of its meaning. During the Renaissance in the West, Girolamo Kadano divided a secret message into words and then embedded these words with certain rules into various positions of a seemingly ordinary written letter. The secret information was then delivered by mail. The above methods can be regarded as the precursors to a technology called information hiding. Today, the rapid development of science and technology has brought about explosive growth in information quantity, followed by problems such as how to ensure the security and integrity of important information and how to protect intellectual property rights. Information hiding technology has emerged to solve these problems, and plays a very important role in modern society [1–3]. The hiding of information can be accomplished using a variety of carriers. Hiding usually does not change the characteristics of the carrier, but uses redundancies in the carrier to complete the information embedding. Therefore, when using this technology, carriers with greater redundancy are typically used, such as audio and video. The corresponding research results are also very rich. Text has been one of the most important and most extensive information dissemination media from the time text was first developed up to the digital age. The huge scale of text usage enables its use as a better carrier of hidden information. However, the lower redundancy associated with text restricts effective information hiding. Therefore, research based on text information hiding is of practical interest.

Text-based information hiding technology has become a research hotspot in the field of information security [4]. The field is mainly divided into format-based, linguistics-based, and coverless text steganography. The first type of method mainly uses the characteristics of the document structure to embed secret information by changing its redundant features. This type of method can be attacked by reorganization, optical character recognition, and other technologies, resulting in the invalidation of the steganographic content. The second type of method achieves secret embedding by changing the structure of sentences while attempting to maintain the meaning of the text. Such a method is linked to many problems, including incoherence in the text semantics, grammatical errors, and sentence breaking, which degrades the readability of the text such that the probability of being detected increases. The third type of method is to construct a language generation model, change the original characteristics of the secret information completely, and then output it to achieve the purpose of covering up the secret. This method is superior to the other two methods in terms of anti-jamming and anti-detection capability and has a high hiding capacity [5,6]. The Markov model and neural networks are often used in nature language generation [7–9]. The former is based on statistical features, while the latter is based on self-learning features. Usually, the latter is more effective, but it has certain drawbacks, such as being difficult to model and requires a long training time. Therefore, the Markov model has an advantage in fast modeling or mobile applications.

In this paper, we propose a coverless text hiding method based on the improved evaluation index and one-bit embedding principle. This principle is based on the Markov model. In the proposed method, we have changed the standard of the previous model. The transition probability of the Markov model is no longer regarded as the only criterion to establish the hiding model. In order to evaluate the quality of the sentences in the training text more objectively and build a better model based on this, we combine the sentence length with the transition probability to complete the follow-up work, which generates better steganographic texts.

The remainder of this paper is arranged as follows. Coverless text steganography and the Markov chain and its application in information hiding are introduced in Section 2. In Section 3, we introduce the methods and specify the implementation steps. In Section 4, we carry out relevant experiments and analyses from various viewpoints to verify the validity of the method. Section 5 discusses the relevant conclusions.

2 Work Basis

2.1 Coverless Text Steganography

In recent years, an increasing number of researchers have begun to pay attention to information hiding methods based on the automatic generation of carriers, that is, coverless text steganography. Early text steganography splits the secret information according to established rules and transforms it into another form, and then embeds these fragments in different positions of the carrier. The secret information is transmitted through the transmission of the carrier. The receiver takes out these fragments, changes them into information, and combines the information together, finally completing the secret information acquisition. Coverless text steganography methods change the traditional way that information hiding must be accomplished by modifying other carriers, and instead directly convert secret information into completely different text. From the perspective of natural language statistical characteristics, the new text generated by the well-designed model will be very close to the training text adopted by the model. Therefore, this method is better at evading malicious attacks and the detection of secret information.

2.2 Markov Chain and Its Information Hiding Application

In engineering science, the Markov chain is an important tool for solving stochastic problems. It was originally proposed by Andre Markov, a Russian mathematician. It consists of a stochastic process with time and state discretization without after-effects. That is, given the current state and all past states of a stochastic process, the conditional probability distribution of its future state depends only on the current state. The sequence of random variables contained in the Markov chain has the following Markov property.

Let $\{X(t), t \in T\}$ be a random process, and E is its state space. For any $t_1 < t_2 < \ldots < t_n < t$, and any $x_1, x_2, \ldots, x_n, x \in E$, the conditional distribution function of random variable $X(t)$ under the known variable $X(t_1) = x_1, \ldots, X(t_n) = x_n$ is only related to $X(t_n) = x_n$, and is completely independent of $X(t_1) = x_1, \ldots, X(t_{n-1}) = x_{n-1}$. That is, the conditional distribution function satisfies the Eq. (1).

$$F(x, t|x_n, x_{n-1}, \ldots, x_2, x_1, t_n, t_{n-1}, \ldots, t_2, t_1) = F(x, t|x_n, t_n) \tag{1}$$

This property is called the Markov property, also known as the no after-effect or memoryless property [10].

The basis of coverless text steganography is natural language generation technology. In this field, natural language generation is usually realized by a language statistical model. The Markov chain is one of the important representatives of this type of model. Therefore, it is often used as the foundation for the coverless steganography model. The essence of the model is reflected in the state transition graph. The graph reflects the distribution between words in the training text used to build the model. Therefore, the statistical characteristics of the text generated by the model are similar to the training text. The authors in [11] proposed a text hiding method and completed the steganography and extraction processes based on the Markov chain. This process is shown in Fig. 1.

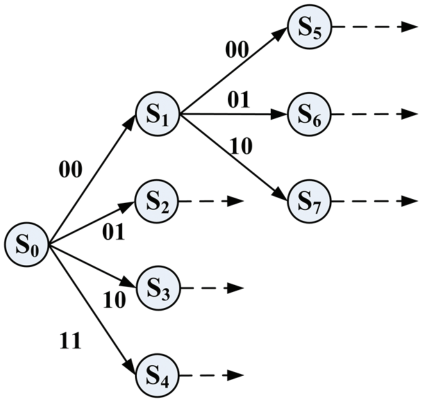

Figure 1: Information steganography and extraction based on a state transition graph

The initial word, “S_0”, was first selected and used to determine the corresponding state transition graph. The secret information converted to binary coding corresponded to the coding in the state transition graph elementwise, and a steganographic sentence beginning with “S_0” was generated. Supposing that the information embedded in each state transition is 2 bit. When the hidden information is 00 10, the text $S_0S_1S_7$ can be obtained according to the state transition graph. The information hiding process is shown in Fig. 2. With this method, the original secret information could be obtained by comparing the steganographic text with the corresponding state transition graph. This process is the reverse process of the steganography process. Suppose the obtained steganography statement is $S_0S_1S_6$. That is, when the current state is S_0, the information 00 is extracted according to its subsequent word or phrase S_1, and information 01 is extracted according to the next word or phrase S_6. The information extraction process is shown in Fig. 3.

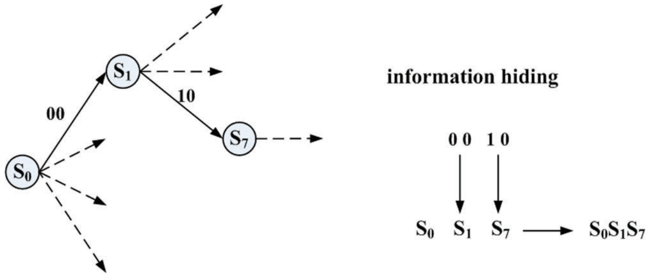

Figure 2: Information hiding

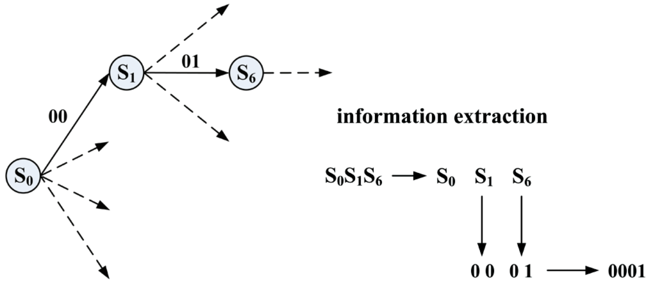

Figure 3: Information extraction

Other studies in [12–14] have evaluated steganography and extraction algorithms linked to various perspectives based on the Markov chain model. Among them, some methods assumed that all transition probabilities were equal. The purpose was to simplify the design and analysis of steganography. However, this neglected the role of the transition probability representing the connection between words in the model. Although this type of algorithm was streamlined, the randomness of the generated sentences was high, which affected the model’s quality. In other approaches, the role of the transition probability was recognized and used in steganography. However, subsequent analyses indicated that the quality of the sentences generated was low and that attacks by statistical detection were possible.

3 Steganography Method Based on Improved Evaluation Index and One-Bit Embedding

3.1 Establishment and Problems in Adjustment of a State Transition Graph

3.1.1 Establishment of State Transition Graph Based on Training Texts

According to the predetermined grammar rules, all sentences are classified according to the same sentence beginning. Sentences with the same first word together form a state transition graph. For

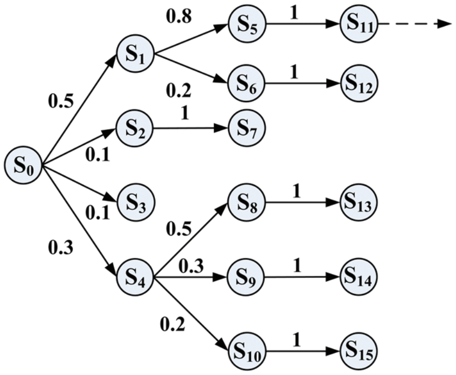

Figure 4: State transition graph corresponding to training texts

example, the sentences beginning with S_0 in the training set were as follows: S_0 S_1 S_5 S_{11}……; S_0 S_1 S_6 S_{12}; S_0 S_2 S_7; S_0 S_3; S_0 S_4 S_8 S_{13}; S_0 S_4 S_9 S_{14}; and S_0 S_4 S_{10} S_{15}. According to these relationships, a state transition graph with S_0 as the start of the sentence was established, and the transition probability in the graph was determined according to the relationships among the words in the sentence, as shown in Fig. 4.

3.1.2 Problems in the Model Building

To build a better model, large-scale texts are needed for training. In the state transition graph formed by such a training set, there will be many branches. Correspondingly, in the process of sentence generation, samples of varying quality should be randomly selected from the training set to guide the generation of steganographic statements.

To generate steganographic text of the highest quality possible, it is necessary to retain the better sentences of the training set in the model. The basis of the selection can be linked to the transition probability of the branches in the state transition graph, reflecting the degree of tightness between words. For words that often appear in the training set elementwise, the transition probability is typically higher. However, using the transition probability as the only basis for the state transition graph adjustment is overly simplistic.

As indicated in Fig. 5, when the transition probability is used as the only factor for model adjustment, if only one optimal branch can be left for the graph, since the connectivity degree of S_0 S_1 is greater than S_0 S_2, it seems more appropriate to retain S_1 branch. However, considering an alternative vantage point, there is only one word or phrase connected in the S_1 direction and four words or phrases connected in the S_2 direction. This indicates that the latter statement is longer. The probability of long sentences is lower than that of short sentences; however, this is acceptable in general texts and does not imply that the quality of a long sentence is necessarily low. Therefore, it is possible to make mistakes by adjusting the state transition graph and guiding and generating sentences based solely on the transition probability. In this paper, we propose a method to adjust the state transition graph using a new perspective.

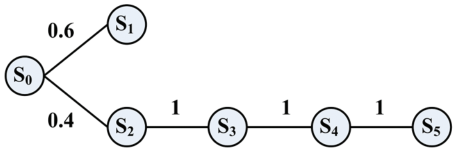

Figure 5: A problem in model adjustment

3.2 Adjustment of State Transition Graph and Information Matching

3.2.1 Graph Adjustment

The state transition graph is adjusted according to two principles:

a)

Let

$$P = a \times b \tag{2}$$

where P is the pruning factor, a is the transition probability of the branch where the node is located, and b is the depth of the tree whose root is the branch node.

b)

We sort the branches of each node from large to small according to the P and retain the first two corresponding branches at most. For Fig. 4, the node S_0 has four branches. According to the evaluation criteria, the four branches are evaluated and sorted, and $P(S_1) > P(S_4) > P(S_2) > P(S_3)$ is obtained. Keep two branches with the highest evaluation factors of S_0, that is, S_1 branch and S_4 branch. This process is shown in Fig. 6. Next, only S_4 does not meet the branch number requirements. According to the evaluation criteria, three branches are evaluated and ranked, and the result is $P(S_8) > P(S_9) > P(S_{10})$. Keep two branches with the highest evaluation factors of S_4, that is, S_8 branch and S_9 branch. This process is shown in Fig. 7. The final adjustment result is shown in Fig. 8.

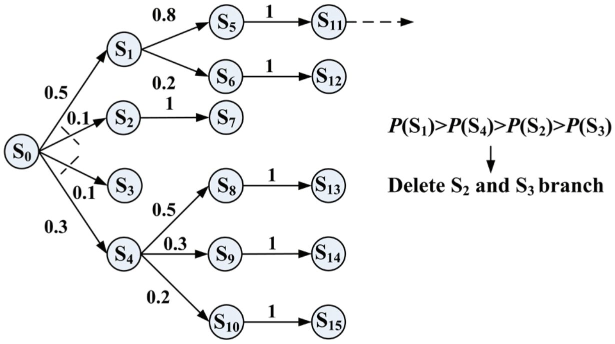

Figure 6: Adjustment of Fig. 4—first adjustment

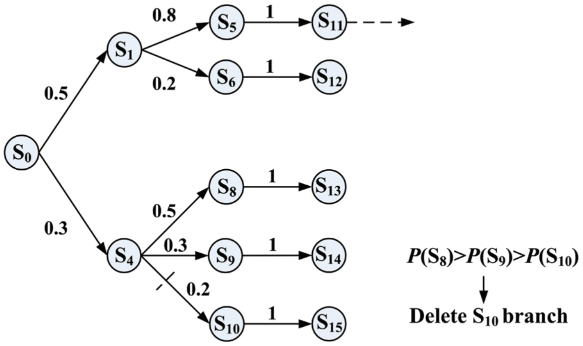

Figure 7: Adjustment of Fig. 4—second adjustment

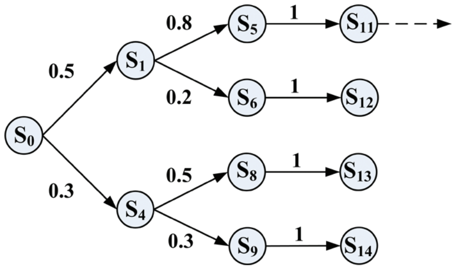

Figure 8: Adjustment of Fig. 4—final result

3.2.2 Information Matching of State Transition Graph

When receivers want to restore the secret data stream correctly after it has been received, they need to identify the information matching rules. In other words, the process of steganography of information should be based on a clear binary meaning conveyed by the graph. As shown in Fig. 9, the probability of an ordered phrase S_0 S_1 is 0.5, and the encoding is 0, while the probability of the word order S_0 S_4 is 0.3, and the encoding is 1. According to the graph, the input secret bit stream is transformed into new text for output. Here is a simple example for steganography. When the model is established according to the design rules, the next step is to wait for receiving the secret information flow and converting it into the new text. Secret bit stream is 0 1 1 0. Give a keyword randomly, which is “she”. Compared with the binary code matched by the model, the data stream will be converted to “she is a lovely girl” and output. The steps for information hiding based on the proposed approach can be expressed as Algorithm 1.

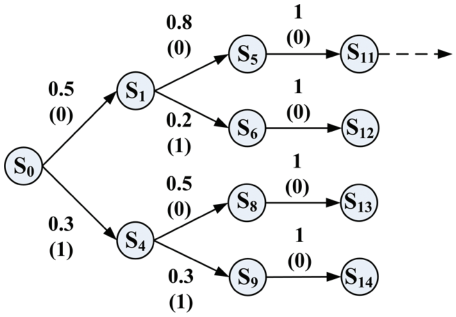

Figure 9: Information matching

3.3 Information Restoration

To recover the secret information correctly, we need to adopt the same model and adjustment rules. When the receiver obtains the new text in the form of steganography, the model and rules are used in reverse processing to recover and obtain the secret information. The extraction process of the former example is as follows. The receiver establishes the model with the same training text and uniform rules. The receiver receives “she is a lovely girl”. Compared the model with the sentence “she is a lovely girl”, the secret bitstream 0 1 1 0 is obtained. The information restoration process is shown in Algorithm 2.

4 Experiment and Discussion

We present and discuss some experiments in this section to analyze the performance of the model objectively from different perspectives.

Algorithm 1 Information Hiding

```
Input:
  Training dataset $D$
  Secret bit stream $B = \{b_1, b_2, \ldots, b_m\}$
  Keyword list $K = \{key_1, key_2, \ldots, key_n\}$
Output:
  Multiple steganography sentences $S = \{s_1, s_2, \ldots, s_p\}$
  Train the dataset and establish a Markov state transition graph;
  Calculate the $P$ value of each node in the Markov state transition graph;
  Adjust the state transition graph according to the definition of $P$, and the two branches with the largest
  $P$-value after each node is retained;
  Match binary coding with the adjusted state transition graph;
    while not the end of $B$ do
      if not the end of the current sentence then
          Match the bit stream with the adjusted state transition graph and generate the
        sentence $s_i$;
      else
          Randomly select a new keyword from $K$ as the start of the new sentence;
    return The steganography sentences generated
```

Algorithm 2 Information Restoration

```
Input:
   Multiple steganography sentences $S = \{s_1, s_2, \ldots, s_p\}$
Output:
  Secret bit stream $B = \{b_1, b_2, \ldots, b_m\}$
  The state transition graph is generated from the same training dataset according to pre-established rules
with the sender;
  Adjust the state transition graph under the same rules;
  Match the binary coding with the state transition graph under the same rules;
    for each sentence $s_i$ do
      Enter the start word $key_j$ of the sentence $s_i$;
      Based on the state transition graph which root is $key_j$, compare it with the steganographic sentence
      and obtain the original input bits;
      Append the original input bits to $B$;
   return Secret bits stream $B$
```

4.1 Data Preparation

The most important function of the coverless text steganography model is to hide secret text. One would hope that these steganographic texts would conform to the characteristics of natural language and be close to the statistical features of the training set. Accordingly, we use a large number of natural texts as a training set for the model. From the Internet, we selected several different types of huge data sets to complete the training and testing of the model. The dataset in [15] was published by Alec Go, which includes a large number of microblogs extracted from Twitter. The dataset in [16] was released by Maas, and contains a large number of movie comments. The dataset in [17] is a news dataset containing 143,000 articles with topics mainly related

Table 1: The training datasets

Dataset	Twitter	IMDB	NEWS
Average Length	9.61	20.01	22.15
Sentence Number	2,598,175	1,195,435	1,919,151
Words Number	24,972,420	23,920,684	42,513,506
Unique Number	45,963	47,813	41,569

to politics. After obtaining the training datasets, we preprocessed them; this involved, among other tasks, case conversion and deletion of special symbols. The processed dataset is described in Tab. 1 in detail.

4.2 Test and Discussion

Because the steganography performance of the model proposed in [9] is better than others, we choose this model to compare the time required for hiding information. In addition, we choose [8] for comparison. This paper described a steganography method based on the Markov model. The method emphasized and completely relied on the transition probability to complete the model. On the basis, in order to complete the steganography of secret information, the method also tried to complete the information embedding under the rule of 1 bit. However, according to the existing analysis in this paper, it is not objective to only use transition probability as the basis of model establishment and sentence quality evaluation. A part of high-quality expression sentences will be deleted by mistake in the process of feature extraction of training text. The experiment used a laptop based on an Intel Core i5-8250u CPU without graphics acceleration. We generated 1,000 sentences with a length limit of 50 words under the condition of 1-bit embedding. Thus, the average embedding time was obtained. As shown in Tab. 2, because our model is relatively simple, the results indicate that the time required for information embedding is less than the embedding time required by a neural network based on [9]. In addition, because the model proposed in the [8] is the simplest, the information embedding time is also the smallest. However, compared with this paper, the embedding time of the two belongs to the same level of magnitude, and the gap is very small.

Table 2: The average embedding time results

Embedded bits(b)	Method in [9](s)	Method in [8](s)	Ours(s)
1	14.185	0.190	0.303

From a natural language processing and generation perspective, we choose relevant indicators to gauge the quality of steganographic sentences, that is, to gauge the similarity between the sentences generated and the training set. The higher the similarity, the better the steganographic quality. The language generation model's quality is often evaluated by *Perplexity*, whose value is inversely proportional to the quality of the model.

$$Perplexity = 2^{-\frac{1}{m}logp(s)} \tag{3}$$

where s is the newly constructed sentence; $p(s)$ is the probability distribution over words in s; and the number of words in the generated sentence is expressed by m.

Based on the index of *Perplexity*, we chose four methods for comparison. Three of them are based on Markov methods, and are described in [8, 12, 13]. The other method is based on a neural network [9].

Table 3: The concealment comparison

Dataset	Twitter	IMDB	News
Method in [12]	441.69 ± 139.94	421.53 ± 123.57	461.38 ± 141.16
Method in [13]	171.77 ± 141.41	200.84 ± 131.53	250.01 ± 131.57
Method in [9]	19.31 ± 5.15	19.93 ± 3.67	17.41 ± 5.96
Method in [8]	18.49 ± 6.31	16.59 ± 7.87	17.19 ± 12.44
Ours	16.78 ± 7.89	14.97 ± 2.55	25.92 ± 18.59

Because two methods based on fixed length code (FLC) and variable length code (VLC) are designed in [9], and the fixed-bit embedding method is used in this paper, the comparison benchmark chosen in the experiments is the FLC method. The results are shown in Tab. 3 and Fig. 10.

As can be seen from Tab. 3 and Fig. 10, although the datasets we choose belong to different categories and have dissimilar linguistic features, the perplexity of our model is generally lower than that of the other models based on the Markov chain. In [12], the author did not emphasize the important role of transition probability when using the Markov chain to build a steganography model. Therefore, the quality of the generated statements was difficult to be guaranteed. In [13], although the author recognized and mentioned the role of transition probability in the designed method, the transition probability was not well utilized in the model, and the steganography effect was not outstanding. In [8], the role of transition probability was emphasized in the model, and it was used as the foundation of the steganography model. However, it was easy to make misjudgment if the transition probability was used as the only evaluation standard of sentence quality. Therefore, the experimental results in the literature are generally worse than the relevant test data based on the new method. Overall, the text generated using the proposed model in this paper is statistically closer to the training set. In addition, neural network-based language generation models usually perform much better on various performance indicators. However, the results of the experiment indicate that the quality of the text generated by the model proposed in this paper is statistically very close to that of the neural network model in [9]. We adopt the simplest method to directly extract the sentences with the highest evaluation quality in the training text as the core guidance for model

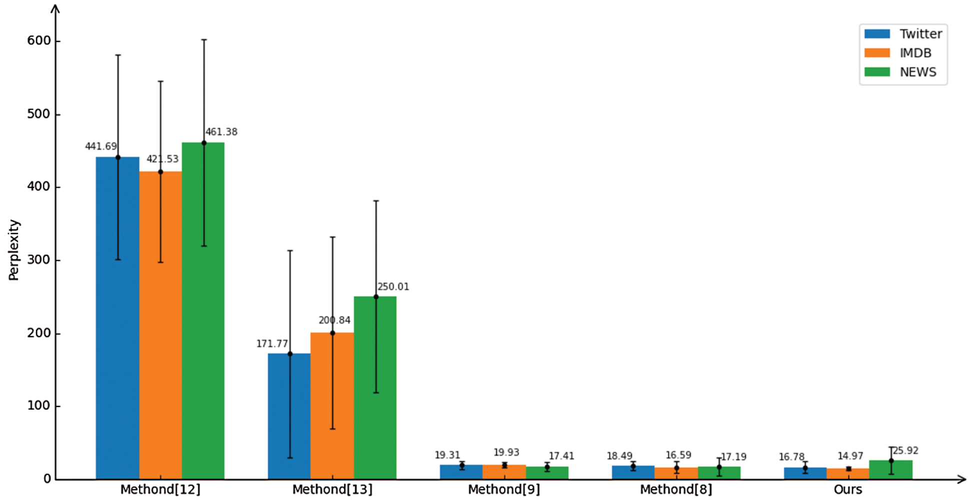

Figure 10: The perplexity of different methods

construction to generate steganographic sentences. Considering the time required and the cost associated with training neural networks, the proposed model appears attractive.

The authors in [18] proposed a method of steganalysis. We can use this method to test the anti-detection capability of the sentences generated by our model. *Accuracy* is an important index used in the method to measure the anti-detection capability. The stronger the resistibility of the model, the closer its resulting value is to 0.5. *Accuracy* is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{4}$$

where *TP* is True Positive, *TN* is True Negative, *FP* is False Positive and *FN* is False Negative. Based on the index, we chose the methods based on [9] and [8] for comparison. For the reasons mentioned above, we still chose the method based on FLC in [9] for comparison. The results are shown in Tab. 4. The test results based on the model proposed in this paper are close to those based on the neural network model [9]. In addition, compared with [8], its performance is better. In general, the method proposed in this paper can be better adapted to different types of data sets. The reason is that in this method, the sentences which can best represent the training text features are extracted as the basis of model construction. For the training text with more uniform language rules, the effect of model in anti steganalysis is better than other methods. For example, the news data set selected in the experiment is very large and has a wide range of sources, but due to the requirements of the news itself, the sentences are very standard and the language features are very similar. On the contrary, for texts with different language styles, such as twitter, the performance will be affected. Although the selected sentences are ranked higher based on the sentence quality and can represent a larger number of training text sentence features, compared with the very different styles and a large number of sentences in the data set, their proportion is still small. Likewise, considering the training cost and the training time of the neural network model, the proposed model is more efficient.

Table 4: The steganalysis comparison

Dataset	Method in [9]	Method in [8]	Ours
Twitter	0.478	0.436	0.455
IMDB	0.460	0.462	0.465
News	0.562	0.582	0.463

The embedding rate (*ER*) is also an important index of the steganography effect, reflecting the amount of embedded information in a certain text. The mathematical expression is

$$ER = \frac{\bar{L} - 1}{8 \times \bar{L} \times \bar{m}} \tag{5}$$

where the $\overline{L}$ indicates the sentence average length in the newly constructed text and the $\overline{m}$ indicates the average number of letters in each word of the new text. Based on the index, we selected several steganography methods to test. The results are shown in Tab. 5 and Fig. 11. In this experiment, the comparison methods we choose are all based on the coverless information hiding method. Compared with [19] and [20], two kinds of steganography methods based on different perspectives, the steganography model proposed in this paper shows better performance. Compared with the methods based on [8] and [9], similar to the analysis in the previous experiments, our method still shows better performance. On the whole, the proposed model in this paper appears to have a higher hiding capacity than the other models.

Table 5: The steganography capacity comparison

Methods	Embedding rate (%)
Method in [19]	1.06
Method in [20]	1.53
Method in [9]	2.48
Method in [8]	2.71
Ours	2.95

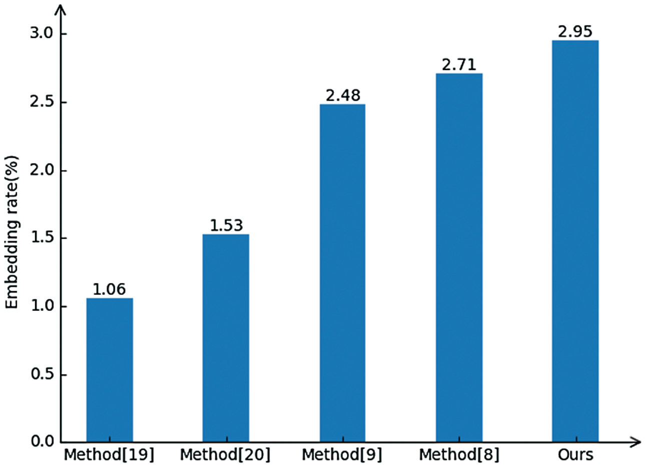

Figure 11: ER of different methods

5 Conclusion

There are two main objectives when hiding information: enhance the transmission security of a secret message using a public channel (high concealment), and increase the amount of secret information that the generated text can carry (high embedding capacity). In line with these two objectives, a new steganographic model based on a Markov chain that is popular in natural language processing is introduced in this paper. Using this model, the quality of the text is not simply based on transition probability. Instead, the sentence length is combined with the transition probability, which leads to a more objective evaluation of the sentence quality and can help to retain high-quality sentences. This method only retains the two sentences with the highest quality, ensuring the steganographic text containing the secret information is generated along a better path. Therefore, the quality of the steganographic text is much better. Because the size of the training set is usually very large, the statements generated by the model will not appear to be excessively repeated. Based on the experimental results, the method proposed in this paper has lower perplexity and a higher embedding rate than other Markov chain-based algorithms. Therefore, it meets the expectations of an information hiding technology that requires high concealment and large capacity. Further, it reduces the probability of detection and is highly secure when the text generated is propagated through public channels. In addition, its performance is very close to that of the neural network. Considering that our model is simpler and the modeling cost is lower, it also has some advantages. In the future, we will study the related algorithms in more detail and we expect to improve this method further.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This work is supported under the National Key Research and Development Program of China (2018YFB1003205), in part under the 2019 Longyuan Youth Innovation and Entrepreneurship Talents (Team) Project (GanZuTongZi [2019] No. 39) (No. 23).

Conflicts of Interests: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Huang, Y. F., Tang, S., Yuan, J. (2011). Steganography in inactive frames of voip streams encoded by source codec. *IEEE Transactions on Information Forensics and Security, 6(2),* 296–306. DOI 10.1109/TIFS.2011.2108649.
2. Huang, Y., Liu, C., Tang, S., Bai, S. (2012). Steganography integration into a low-bit rate speech codec. *IEEE Transactions on Information Forensics and Security, 7(6),* 1865–1875. DOI 10.1109/TIFS.2012.2218599.
3. Xiao, B., Huang, Y. F. (2008). Modeling and optimizing of the information hiding communication system over streaming media. *Journal of Xidian University, 35(3),* 554–558.
4. Luo, Y., Huang, Y. (2017). Text steganography with high embedding rate: using recurrent neural networks to generate chinese classic poetry. *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security,* pp. 99–104.
5. Zhou, Z., Cao, Y., Sun, X. (2016). Coverless information hiding based on bag-of-words model of image. *Journal of Applied Sciences, 34(5),* 527–536.
6. Wu, Y., Sun, X. (2018). Text coverless information hiding method based on hybrid tags. *Journal of Internet Technology, 19(3),* 649–655.
7. Luo, Y., Huang, Y., Li, F., Chang, C. (2016). Text steganography based on ci-poetry generation using Markov chain model. *TIIS, 10(9),* 4568–4584.
8. Wu, N., Shang, P., Fan, J., Yang, Z., Ma, W. et al. (2019). Research on coverless text steganography based on single bit rules. *Journal of Physics: Conference Series, 1237(2),* 022077, IOP Publishing.
9. Yang, Z. L., Guo, X. Q., Chen, Z. M., Huang, Y. F., Zhang, Y. J. (2018). RNN-stega: linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security, 14(5),* 1280–1295. DOI 10.1109/TIFS.2018.2871746.
10. Whittaker, J. A., Thomason, M. G. (1994). A Markov chain model for statistical software testing. *IEEE Transactions on Software Engineering, 20(10),* 812–824. DOI 10.1109/32.328991.
11. Wu, S. F. (2003). *Researches on information hiding technology.* China: College of Computer Science and Technology. USTC.
12. Dai, W., Yu, Y., Dai, Y., Deng, B. (2010). Text steganography system using Markov chain source model and des algorithm. *JSW, 5(7),* 785–792.
13. Moraldo, H. H. (2014). An approach for text steganography based on Markov chains. *arXiv preprint arXiv: 1409.0915.*
14. Shniperov, A., Nikitina, K. (2016). A text steganography method based on Markov chains. *Automatic Control and Computer Sciences, 50(8),* 802–808. DOI 10.3103/S0146411616080174.
15. Go, A., Bhayani, R., Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1(12).*
16. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. et al. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies,* pp. 142–150.
17. Thomson, A. (2005). News dataset. https://www.kaggle.com/snapcrack/all-the-news/data.

18. Meng, P., Hang, L., Yang, W., Chen, Z., Zheng, H. (2009). Linguistic steganography detection algorithm using statistical language model. *International Conference on Information Technology and Computer Science,* vol. 2, pp. 540–543. IEEE.
19. Chen, X., Sun, H., Tobe, Y., Zhou, Z., Sun, X. (2015). Coverless information hiding method based on the Chinese mathematical expression. *International Conference on Cloud Computing and Security,* pp. 133–143. Springer.
20. Zhou, Z., Mu, Y., Zhao, N., Wu, Q. J., Yang, C. N. (2016). Coverless information hiding method based on multi-keywords. *International Conference on Cloud Computing and Security,* pp. 39–47. Springer.