STBS-Stega: Coverless text steganography based on state transition-binary sequence

Ning Wu1,2, Zhongliang Yang3, Yi Yang1, Lian Li1, Poli Shang4, Weibo Ma5 and Zhenru Liu5

Abstract
Information-hiding technology has recently developed into an area of significant interest in the field of information security. As one of the primary carriers in steganography, it is difficult to hide information in texts because there is insufficient information redundancy. Traditional text steganography methods are generally not robust or secure. Based on the Markov chain model, a new text steganography approach is proposed that focuses on transition probability, one of the most important concepts of the Markov chain model. We created a state transition-binary sequence diagrams based on the aforementioned concepts and used them to guide the generation of new texts with embedded secret information. Compared to other related works, the proposed method exploits the use of the transition probability in the process of steganographic text generation. The associated developed algorithm also encrypts the serial number of the state transition-binary sequence diagram needed by the receiver to extract the information, which further enhances the security of the steganography information. Experiments were designed to evaluate the proposed model. The results revealed that the model had higher concealment and hidden capacity compared to previous methods.

Keywords
STBS-Stega, coverless text steganography, Markov chain model, transition probability, state transition-binary sequence diagram

Date received: 30 September 2019; accepted: 27 February 2020
Handling Editor: Yunpeng Li

Introduction
Information hiding is a covert transmission technique occurring via a public channel by hiding meaningful information in non-secret carriers.1–4 It is widely used in data secrecy communication, identity authentication, copyright protection of digital works, tracking of piracy and integrity, authenticity identification, content recovery, and so on.

Information hiding usually needs a carrier. Text is widely used as an important medium but it is difficult to use it as an information-hiding carrier because of its low redundancy.5,6 For the reason, the research results of text-based information-hiding technology are much

---

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en/us/nam/open-access-at-sage).
less than those of image, video, and other multimedia, so the significance of the research is become more prominent. Previous works on text information hiding could be summarized as follows:

a. **Text information hiding based on the format of the carrier.** This type of method uses the attribute of text to hide information. Usually, it can be divided into the following methods. The first method is realized by changing the line spacing, word spacing, or character feature coding of text. The second method is realized by changing character features, such as size and color. The third method is realized by adding or deleting some invisible characters in the text, such as space and tabs. The above methods are easy to implement, but the robustness and security are not high.

b. **Text information hiding based on the content of the carrier.** This method could be divided into two categories: carrier modification–based methods and carrier automatic generation–based methods.

The former methods include the method based on synonym substitution and the method based on syntax or semantics. The synonym substitution method is based on the previously established synonym library, replacing the words of the carrier text with the synonyms to realize the purpose of information hiding. However, for complex languages (such as Chinese), synonym substitution needs to consider the context. This greatly increases the difficulty of word substitution. The syntax-based method refers to changing the sentence structure or adding redundant words to achieve the purpose of hiding information while maintaining the original meaning. The method has a high steganography capacity. However, because sentence generation is usually done by machine, the generated text is poor at the semantic level and easy to be detected manually. The semantic-based method mainly relies on semantic analysis of text. The best way is to use ontology theory. However, because the process of building the ontology library is complex, the implementation of the method is difficult.

The implementation of the latter method does not need a carrier. The information that needs to be hidden will be directly transmitted after processed by a language generation algorithm that conforms to the statistical characteristics of natural language. This method can avoid the steganalysis based on statistical features better. It includes two methods. One is based on the Markov model; the other is based on the neural network model.

Markov model is a classical statistical model, which has many applications in natural language processing. The neural network model imitates the functional characteristics of the human nervous system. It has the ability of large-scale parallel, distributed storage, and self-learning. It is also widely used in natural language processing. Generally speaking, this method has a better modeling effect. However, compared with the Markov model, there are some shortcomings, such as the need for better hardware support, longer model training time, and loading time. Therefore, the Markov model has more obvious advantages in situations where rapid modeling is required, or steganography in general hardware environments, and similar applications.

This study exploits the Markov chain model to propose a novel coverless text steganography method called state transition-binary sequence (STBS)-based steganography methodology (STBS-Stega). This approach allows a secret message to be passed through to generate semantically smooth texts. In the process of text generation, we emphasize the importance of the transition probability and extensively utilize transition probability to ensure that the generated texts are more smooth and natural. The concealment and hidden capacity of this model are superior to previous models.

The remainder of this article is organized as follows. Section “Related work” introduces related works. Section “STBS-Stega methodology” introduces the framework and details of the STBS-Stega algorithm. Section “Experiments and evaluation” presents the experimental evaluation results and discussion. Finally, the main conclusions are summarized in section “Conclusion.”

**Related work**

Steganography methods based on the automatic generation of carriers have become a hot topic in recent years. In the methods, secret information is put into a mathematical model that conforms to the statistical characteristics of natural language, and then the new text is generated and transmitted to achieve the purpose of hidden communication. The method proposed in this article belongs to them. In this section, we first introduce the related technical background of the algorithm, that is, the Markov chain model. Then, we introduce the application of the Markov chain model in steganography.

**The Markov chain model**

The Markov chain, named after Andre Markov, is a discrete random process. It describes a state sequence. It consists of a random process with time and state discretization, without aftereffects. That is to say, given the current state, the past state of the process is
ineffective in predicting the future state of the current. In every step of the Markov chain, the system can change from one state to another, or it can also maintain the current state, according to the probability distribution.

The sequence of random variables contained in the Markov chain exhibits the Markov property. The range of these variables, that is, the set of all their possible values, is called the “state space.” The value of \( X_n \) is the state of time \( n \). If the conditional probability distribution of \( X_{n+1} \) to the past states is only a function of \( X_n \), equation (1) is given as follows

\[
P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n)
\]

where \( x \) is a state in the process. This equation can be regarded as the Markov property.\(^7\)

**Application of steganography**

The traditional information-hiding communication methods are to split the secret information in a specific way, and then embed the fragmented information in different locations of the carrier, and finally send the carrier with the secret information through the public channel. The receiver extracts the fragmented secrets through the reverse process and combines them to obtain the complete content. If the carrier is found by the enemy, the hidden information will be easily detected and extracted according to known information embedding rules and analysis methods. In addition, the enemy can attack the carrier containing secret information through some technical means to destroy the integrity of the hidden information.

The information-hiding method based on the automatic generation of the carrier does not need other carriers but directly transforms the secret information into the carrier according to the statistical characteristics of natural language. In other words, the carrier is the hidden information, and the hidden information is the carrier. Therefore, it is difficult to detect steganography using steganalysis based on natural language statistical characteristics or other general steganalysis methods. Therefore, the method is more secure.

The basis of the steganography method is natural language processing and generation. In the field of natural language processing, the statistical language model is widely used to model a text. The Markov chain model could be used as a better approximation of the statistical language model.\(^8\)–\(^10\) Using this model, it is easy to generate texts that are similar to natural language. Therefore, the model is often used for steganography and the associated methods are called coverless text steganography. The state transition diagram is the core of the model. It is based on the assumption of the N meta-grammatical model, which describes the distribution rules between words obtained from statistics in a large number of natural language texts.

Previous literature\(^11\) described the state transition diagram and the process of hiding and extracting information. Given a start state, subsequent state transitions were assigned binary information. Information hiding and extraction were based on the matched binary information, as shown in Figure 1.

The process of steganography can be described as follows. The start state is given as \( S_0 \). Supposing that the information embedded in each state transition is 2 bits, if the secret information is 01, the text \( S_0S_2S_8 \) could be obtained according to the state transition diagram.

The process of extraction can be described as follows. This is the reverse process of the steganography process. That is, when the current state is \( S_0 \), the information 01 is extracted according to its subsequent word or phrase \( S_2 \), and information 11 is extracted according to the next word or phrase \( S_8 \).

An important concept in the state transition diagram is the transition probability, which indicates the probability of changing to the next specific state. Previous literature\(^12\)–\(^15\) designed the steganography method based on Markov chain models from different perspectives. In order to simplify the generation process of natural sentences and related design and analysis, some methods assumed that the transition probability from a given state to other states was equal. These methods ignored the relationship between words in the training text, so the new text generated by the model could not reflect the characteristics of the training text, which caused the quality of the sentence to be generated not very high. In other methods, the existence of transition probability was emphasized, but through analysis, the
quality of statement generation had not been greatly improved.

**STBS-Stega methodology**

To maximize the quality of the generated natural texts based on the training text and to avoid the steganalysis based on statistical characteristics, we focused on the concept of transition probability in a Markov chain and designed the steganography model based on it. This section will introduce the implementation of this algorithm in detail.

**The introduction of the transition probability**

Figure 2 is the state transition diagram containing transition probabilities. The transition probability from statistics indicates the possible degree from a preceding word or phrase to a subsequent one in the training text. According to the statistics of the training text, the transition probabilities of a word or phrase $S_0$ to four words or phrases $S_1$, $S_2$, $S_3$, and $S_4$ are 0.3, 0.4, 0.2, and 0.1, respectively. The transition probability of $S_1$ to $S_5$ is 1, and the transition probabilities of $S_2$ to $S_0$, $S_7$, $S_8$, and $S_9$ are 0.3, 0.2, 0.1, and 0.4, respectively, and so on. The transition probabilities of a word or phrase $S_0$ to subsequent words or phrases are $S_2$, $S_1$, $S_3$, and $S_4$ in descending order. This indicates that the probability of the appearance of the phrase $S_0S_2$ in the training text is greater than $S_0S_1$ and so on.

**The steganography process and the problem of using the transition probability**

**Frequency sorting.** According to Figure 2, a statement is generated from the start word or phrase $S_0$. If one wants to obtain an optimal sentence, the transition probability should be considered for the selection of the next word or phrase. The phrase with the highest probability is $S_0S_2$, thus forming phrase $S_0S_2$. Starting from the word or phrase $S_2$, the next phrase is also selected according to probability. $S_2S_9$ has the highest probability, so the phrase $S_0S_2S_9$ is formed. The text is generated in turn. The sorting and text generation processes are shown in Figure 3.

Consider the following situation:

*The first time:* the sender $S$ sends a text sequence 01 10 11 ... to the receiver $R$. According to the priority rule of transition probability, the generated text sequence is $S_0S_2S_9S_i$...

*The second time:* the sender $S$ sends a text sequence 11 01 10 ... to the receiver $R$. According to the priority rule of transition probability, the generated text sequence is $S_0S_2S_9S_i$...

*The third time:* the sender $S$ sends a text sequence 10 01 00 ... to the receiver $R$. According to the priority rule of transition probability, the generated text sequence is $S_0S_2S_9S_i$...

Obviously, no matter what text the sender sends, the receiver will receive the same sequence of word based on the principle of optimal sentence generation. This means that the recipient is unable to restore the correct secret message sent by the sender using only the Markov state transition diagram generated from the training text and the received binary sequence. Therefore, by simply introducing the transition probability and using the principle of probability priority, although better natural statements can be generated, the hidden text information cannot be correctly decoded.

**Adding information matching rules.** If the receiver wants to decode the hiding information correctly, it is necessary to determine the matching rules from the beginning. Therefore, we should consider not only the transition

---

**Figure 2.** State transition diagram with state transition probability.

**Figure 3.** Transition probability ranking and text generation.
probability but also the explicit binary meaning. As depicted in Figure 4, the probability of phrase \(S_0 \rightarrow S_1\) is 0.3 and the encoding is 01; the probability of phrase \(S_0 \rightarrow S_2\) is 0.4 and the encoding is 00; and so on.

If we simply apply a binary meaning to a branch with a certain probability of transition, the following problems may arise. Assuming that the input stream guided by \(S_0\) is 00 01, the phrase generated is \(S_0 \rightarrow S_2 \rightarrow S_6\). It is determined that for phrase \(S_0 \rightarrow S_2\), the probability of transition is 0.4, the highest value. However, for the sequence \(S_2 \rightarrow S_6\), the probability is 0.3. The word or phrase \(S_9\) with the highest transition probability of 0.4 is not chosen. Therefore, by simply combining the transition probability with a specific binary encoding, the steganography effect is consistent with a negligible transition probability.

A new steganography algorithm combined with transition probability

To avoid the aforementioned problems, a new method named STBS-Stega is proposed.

Preparatory work. According to the selected training text, we read and filtered all the sentences using the same word or phrase as the head of the sentence. This was done in preparation for the subsequent forming of the corresponding state transition diagram. For example, we identified all sentences with \(S_0\) as the start word or phrase in the training text as follows: \(S_0 \rightarrow S_1 \ldots \); \(S_0 \rightarrow S_2 \rightarrow S_6 \ldots \); \(S_0 \rightarrow S_2 \rightarrow S_7 \ldots \); \(S_0 \rightarrow S_2 \rightarrow S_8 \ldots \); \(S_0 \rightarrow S_2 \rightarrow S_9 \ldots \); \(S_0 \rightarrow S_3 \ldots \); \(S_0 \rightarrow S_4 \ldots \).

The establishment of the state transition diagram. The Markov chain state transition diagram with \(S_0\) as the start word or phrase was established and the corresponding weights were determined according to the relationship between the words or phrases, as shown in Figure 5.

When the word or phrase \(S_0\) is at the beginning of the sentence, there are four adjacent words or phrases, namely \(S_1, S_2, S_3,\) and \(S_4\). According to the frequency of each phrase, the corresponding transition probability is 0.29, 0.44, 0.26, and 0.01, respectively. The adjacency word or phrase of \(S_1\) is the only \(S_2\), so the corresponding transition probability is 1. There are four words or phrases adjacent to \(S_2\), which are \(S_0, S_7, S_8,\) and \(S_9\). According to the relation of each word or phrase, the corresponding transition probability is 0.30, 0.22, 0.10, and 0.38, respectively.

The matching of the state transition diagram and binary sequence. To restore the text information after the steganography to the correct original input information, the coding meaning of each state transition branch should be given based on the state transition diagram. The same state transition diagram can form different STBS diagrams because their state transition branch can be matched with different binary information. Each graph gives a fixed serial number. Finally, a series of state transition sequence diagrams with the same shape but different branch-binary sequence matching schemes are formed. In the state transition diagram shown in Figure 5, it is assumed that the maximum branch of each node is four, so two-bit binary numbers can be used to represent the corresponding state transition branch of each word or phrase.

Figure 6 describes four STBS diagrams that are formed from the combination of the branches of the state transition diagram in Figure 5 and the different two-bit binary sequences. Suppose the serial numbers of the four subfigures are 1, 2, 3, and 4.

The differences between the four graphs are as follows:
In Figure 6(a), the $S_2S_6$ branch is represented by binary sequence 01, the $S_2S_7$ branch is represented by binary sequence 10, the $S_2S_8$ branch is represented by binary sequence 11, and the $S_2S_9$ branch is represented by binary sequence 00.

In Figure 6(b), the $S_2S_6$ branch is represented by binary sequence 01, the $S_2S_7$ branch is represented by binary sequence 11, and the $S_2S_9$ branch is represented by binary sequence 00.

In Figure 6(c), the $S_2S_6$ branch is represented by binary sequence 01, the $S_2S_7$ branch is represented by binary sequence 11, and the $S_2S_9$ branch is represented by binary sequence 00.

In Figure 6(d), the $S_2S_6$ branch is represented by binary sequence 01, the $S_2S_7$ branch is represented by binary sequence 00, the $S_2S_8$ branch is represented by binary sequence 00, and the $S_2S_9$ branch is represented by binary sequence 11.

**Input sequence analysis.** To generate superior quality natural sentences, it is necessary to match the information that is prepared for steganography with the state transition diagrams that can form the optimal statement. According to Figure 6, suppose that the sequence of inputs required for steganography is as follows: 00 11 ...

By finding the STBS diagram, the steganographic text generated from the state transition diagram Figure 6(d) is the optimal statement. Therefore, Figure 6(d) is used as a state transition diagram that matches the input sequence. In addition, serial number 4 is transmitted to the receiver of the steganographic information. Correspondingly, the resulting steganographic text is $S_0S_2S_0$ ...

In order to generate more natural steganographic text, we usually choose large-scale training texts. Correspondingly, the set of the start word, keyword, is also very large. In steganography, because the start word is given randomly, the probability of choosing the same start word is limited, that is to say, the probability of generating the same sentence is not high. If we want to further reduce the possibility of this case, we can count the frequency of the use of the start word and set priorities. For words that have reached a certain frequency of use, their priority will be reduced to avoid being used again quickly. In addition, when steganography is about to be completed, there may be a problem that the last secret binary numbers corresponds to multiple state transition diagrams and the serial number could not be determined. In this case, the state transition diagram we choose is the one with the smallest serial number under the same structure.

**The encrypted transmission of the serial number of the STBS diagram.** To ensure the security of steganography information, data encryption standard (DES) and other encryption algorithms can be used to encrypt the serial number of the STBS diagram prior to transmission.

The details of information hiding will be shown in Algorithm 1.

**Algorithm 1. Information Hiding Algorithm.**

**Input:**
Training dataset $D$
Secret bit stream $B = \{b_1, b_2, ..., b_n\}$
Keyword list $K = \{key_1, key_2, ..., key_s\}$

**Output:**
Multiple steganography sentences $S = \{s_1, s_2, ..., s_p\}$
Encrypt serial number set $N' = \{n'_1, n'_2, ..., n'_q\}$

Train the dataset and establish a Markov state transition diagram;
Establish state transition-binary sequence (STBS) diagram based on the Markov state transition diagram and assign serial numbers to them;
while not the end of $B$ do
    if not the end of the current sentence then
        Match the bit stream with the state transition-binary sequence (STBS) diagram that can generate the optimal sentence $s_k$;
        Let $n_i$ equal to the serial number of the diagram;
        Output the $n'_i = encrypt n_i$;
    else
        Randomly select a new keyword from $K$ as the start of the new sentence;
return the steganography sentences generated
The flowchart of the Algorithm 1 is shown in Figure 7. Here is a simple example.

Secret bit stream is 00 10 01 00. Give a keyword randomly, which is “he.” The best sentence starting with “he” is “he is a good boy” in training text. The serial number of the STBS diagram that matches “he is a good boy” with 00 10 01 00 is “5.” So, the output sentence is “he is a good boy” and the serial number “5” is transmitted encrypted.

**Extraction of steganography information**

To restore steganographic information correctly, the sender and receiver must use the same state transition diagram. Prior to the communication between the two sides, a unified training text was established and unified rules were established for the ordering of the STBS diagram. Therefore, when the steganography recipient obtains the encrypted serial number, after decryption, the serial number of the STBS diagram is obtained. After comparison with the steganographic information, the original secret information is extracted.

The extraction process for the example in the previous subsection is as follows.

The receiver establishes an STBS diagram with the same training text and uniform rules. The receiver receives “he is a good boy” and the serial number 5. Find the identical STBS diagram according to “he is a good boy” and the serial number. Finally, compare the optimal sentence “he is a good boy” to extract the secret bit stream 00 10 01 00.

In a limited training sample, the probability is usually very low when the first word is different and the sequence of all the words behind is identical. If the start word is missing, the remaining word sequence of the sentence is compared with all the optimal sentences, and the optimal sentence with the highest matching ratio is selected. In this way, the correct STBS diagram can still be found with great probability, and the original transmitted information is extracted by combining the received serial number.

In the above example, if “he” is lost, it is easier to confirm the STBS diagrams to which it belongs according to “is a good boy” compared with the optimal sentence formed in the training text. The original transmission information can be extracted according to the serial number “5.”

The details of the information extraction process are shown in Algorithm 2.

**Experiments and evaluation**

In this section, we designed several experiments to test the proposed model from the perspectives of concealment and hidden capacity. For concealment, we compared and analyzed the quality of the texts generated at different embedding rates (ERs) with the training text. In addition, we tested and compared the steganography efficiency of different methods. We also used several steganalysis methods to test the antisteganalysis ability of different methods. Finally, in order to test the hidden capacity, we analyzed how much information could be embedded in the generated text and compared it with other text steganography algorithms.
Data preparing

A significant number of human-written natural texts were required to train the proposed model to become a good enough language model because we expect this approach to be able to automatically imitate and learn sentences written by humans. We chose three large-scale text datasets that contained the most common text media on the Internet as the training sets, which were microblog,16 movie reviews,17 and news.18

The first dataset was published by Alec Go et al.16 and includes 1,600,000 sentimental tweets sentences. The second dataset published by Maas et al.17 includes movie reviews from Internet Movie Database (IMDb). Finally, the third dataset is a news dataset18 that contains 143,000 more standard articles. These datasets were preprocessed before training, and the details are shown in Table 1.6

Experimental results and discussion

We use preprocessed datasets to train the model. The example sentences generated from the model are as follows:

well i’m off to bed
enjoy your day

hope i can get a new one
we love you
hope you have a great day
i’m really sorry

Then, we will use some indexes to measure the performance of the algorithm.

Concealment analysis. The concealment of information is mainly reflected in the quality of the generated texts. The better the quality of the text, the higher the concealment. Referring to other related natural language processing works, we used perplexity to evaluate the quality of the generated sentences. In natural language processing, perplexity is a standard way of evaluating language models; the higher the value of perplexity, the worse the language model.

\[
\text{perplexity} = 2^{-\frac{1}{m}\sum \log p(s)}
\]  

where \(s\) is the generated sentence, \(p(s)\) indicates the probability distribution for words in \(s\), and the probability is calculated from the language model of the training texts. Parameter \(m\) is the total number of words in \(s\).6 The higher the perplexity, the greater the

Table 1. The training datasets.

| Dataset      | Twitter      | IMDb        | News        |
|--------------|--------------|-------------|-------------|
| Average length | 9.68         | 19.94       | 22.24       |
| Sentence number | 2,639,290    | 1,283,813   | 1,962,040   |
| Words number | 25,551,044   | 25,601,794  | 43,626,829  |
| Unique number | 46,341        | 48,342      | 42,745      |

IMDb: Internet Movie Database.
statistical distribution difference between the generated text and the training text.

In order to test the performance of the proposed method, we selected three coverless text steganography methods to train the model on three datasets and compared them. The two steganography models in Dai et al.\textsuperscript{12} and Hernan Moraldo\textsuperscript{14} are similar to the model proposed in this article, both of which are based on the Markov chain. The model in Yang et al.\textsuperscript{6} is based on the neural network. Because our method is based on fixed bits, in order to compare on the same basis, the following involves the experimental data in the previous literature,\textsuperscript{6} we all choose the data which are completed under fixed bits embedding. The mean and standard deviation of the perplexity is tested, and the results are shown in Table 2 and Figure 8.

From Table 2 and Figure 8, it is evident that although each dataset belongs to different fields and has the different textual content, the perplexity of the steganographic text generated by the proposed model for each dataset is the smallest compared to the similar models such as Dai et al.\textsuperscript{12} and Hernan Moraldo.\textsuperscript{14} The reason is that the model highlights the role of transition probability. In other words, the most frequently used word sequences in the training text are extracted as the basis of the model, and these sentences reflect the core features of the training text. Therefore, from a statistical point of view, the steganography model proposed in this article is closer to the natural language features of the training text, and the effect is better. The test results of the proposed model are close to the test results based on the neural network model. This advantage is more obvious in situations where rapid modeling is required and hardware conditions are limited. Considering the training costs and the training time of the neural network model, it is deemed that the proposed model is more beneficial. However, it needs to be acknowledged that models based on neural networks, especially models based on complex neural networks, can indeed reflect the special and general characteristics of training texts more objectively. In other words, the generated sentences through related models are richer and more diverse.

In addition, we used the techniques from Meng et al.\textsuperscript{19} and Samanta et al.\textsuperscript{20} to test the steganalysis resistibility of each model, with measurements of accuracy being the result of these tests. The stronger the resistibility of the model, the closer its resulting value is to 0.5. Accuracy represents the proportion that the classifier judges correctly for the entire sample. It is defined as follows

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$ \hspace{1cm} (3)

where $TP$ are true positives, $TN$ are true negatives, $FP$ is false positives, and $FN$ is false negatives. We cite a set of data used by the co-author in Yang et al.\textsuperscript{6} for comparison. Two neural network models are compared. The results of the tests are shown in Table 3. The test results of the model proposed in the article are close to the neural network model. Some results are slightly weaker than those of the neural network. Similar to the conclusion in the previous experiment, considering the advantages of neural networks, such results are acceptable. We can also find that the test results in the News are better. The sentences in News are more precise, and the grammatical structure is more uniform. Therefore, the method proposed in the article is more successful in

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c|c|}
\hline
Dataset & Twitter & IMDb & News \\
\hline
Baseline\textsuperscript{12} & 430.85 ± 144.67 & 418.70 ± 105.32 & 470.54 ± 122.73 \\
Baseline\textsuperscript{14} & 149.22 ± 135.44 & 161.92 ± 143.31 & 175.42 ± 126.28 \\
Baseline\textsuperscript{6} & 19.29 ± 4.39 & 20.17 ± 3.95 & 17.47 ± 6.83 \\
Ours & 14.07 ± 8.83 & 13.34 ± 9.90 & 12.89 ± 8.75 \\
\hline
\end{tabular}
\caption{The mean and standard deviation of the perplexity of the algorithms.}
\end{table}

IMDb: Internet Movie Database.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure8.png}
\caption{The perplexity of different algorithms.}
\end{figure}
extracting the natural language features of the training text, and the generated sentences are closer to the training text.

**Hiding efficiency.** Next, the hiding efficiency of the model is calculated. From the perspective of information hiding, because the hidden model based on previous literature has a better hiding effect, we choose this model to compare the difference of the time needed to hide information. We generate 1000 sentences under different bit embedding and limit each sentence to 50 words. The average embedding time is obtained. The experimental platform uses a laptop without graphics acceleration based on Intel Core i5-8250u. The results are shown in Table 4.

Two conclusions can be drawn from Table 4. First, as the number of embedded bits increases, the information embedding time required by the two models increase in general. In addition, the time required for the proposed model under different embedding bits is less than that based on the neural network model. The reason for this conclusion is that our model is relatively simple, so it does not take too much time to complete the information embedding.

**Hidden capacity.** The ER is also an important index for the evaluation of a steganography algorithm. It describes the amount of information embedding in text. Usually, the quality of the generated text decreases as the amount of information embedded in the text increases. Previous algorithms have shown that it is difficult to simultaneously guarantee concealment and hidden capacity.

The mathematical expression of ER is as follows

\[
ER = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i - 1}{B(s_i)} = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i - 1}{8 \times \sum_{j=1}^{L_i} m_{i,j}} = \frac{\bar{L} - 1}{8 \times L \times \bar{m}}
\]

where \(N\) is the number of generated sentences and \(L_i\) is the length of the \(i\)th sentence. \(B(s_i)\) represents the number of bits occupied by the \(i\)th sentence on the computer. Given that each English letter actually occupies one byte in the computer’s memory, that is, 8 bits, the number of bits occupied by each English sentence is \(B(s_i) = 8\) times the sum from \(j = 1\) to \(L_i\) of \(m\{i,j\}\), where \(m\{i,j\}\) represents the number of letters contained in the \(j\)th word of the \(i\)th sentence. Overline \(L\) and the overline \(m\) represent the average length of each sentence in the generated text and the average number of letters contained in each word, respectively. We also cite a set of data used by co-authors in Yang et al. to compare. It includes different types of text steganography methods. The results are shown in Table 5 and Figure 9.
From the results, it is evident that the proposed model has a steganography capacity that is superior to the other text steganography models. Combining with the previous experiments, it can be proved that the proposed model can achieve relative high concealment and high hiding capacity at the same time.

Conclusion

In this article, we used the Markov model to generate natural language to realize steganography. The past methods based on this model did not fully use an important concept in the Markov chain, that is, the transition probability. As a result, the quality of the steganography is more or less affected. In our research, the transition probability is the key aspect in the generation of steganography based on the Markov chain. We match the input sequence with the most suitable STBS diagram. An optimal sentence is formed along the direction of the maximum transition probability in STBS diagram to achieve a better steganography effect. Based on related experiments, it is proven that the steganography effect of the method is superior to that of other approximate methods in many aspects. The effectiveness of the method is verified. Compared with the neural network–based steganography method, its modeling time and embedded data time are smaller. Also, the hardware environment requirements are lower. For larger training texts, especially those with stricter grammar, the advantages of this method are obvious. On the contrary, for training texts with diversified language styles or training texts with a limited scale, the weakness of this method will appear, such as the single expression form of text or repeated sentences. In future research, we will further explore the method based on the proposed approach and we expect to improve the algorithm.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported under the National Key Research and Development Program of China (2018YFB1003205), in part under the Natural Science Foundation of Gansu (1506RJA057), and in part under the 2019 Longyuan Youth Innovation and Entrepreneurship Talents (Team) Project (GanZuTongZi [2019]No. 39) (No. 23).

ORCID iD

Ning Wu https://orcid.org/0000-0002-4050-0937

References

1. Petitcolas FAP, Anderson RJ and Kuhn MG. Information hiding—a survey. P IEEE 1999; 87(7): 1062–1078.
2. Huang YF, Tang S and Yuan J. Steganography in inactive frames of VoIP streams encoded by source codec. IEEE T Inf Foren Sec 2011; 6(2): 296–306.
3. Huang Y, Liu C, Tang S, et al. Steganography integration into a low-bit rate speech codec. *IEEE T Inf Foren Sec* 2012; 7(6): 1865–1875.

4. Xiao B and Huang Y. Modeling and optimizing of the information hiding communication system over streaming media. *J Xidian Univ* 2008; 35(3): 554–558.

5. Luo Y and Huang Y. Text steganography with high embedding rate: using recurrent neural networks to generate Chinese classic poetry. In: *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, Philadelphia, PA, 20–22 June 2017, pp.99–104. New York: ACM.

6. Yang Z-L, Guo X-Q, Chen Z-M, et al. RNN-Stega: linguistic steganography based on recurrent neural networks. *IEEE T Inf Foren Sec* 2018; 14(5): 1280–1295.

7. Whittaker JA and Thomason MG. A Markov chain model for statistical software testing. *IEEE T Software Eng* 1994; 20(10): 812–824.

8. Shniperov AN and Nikitina KA. A text steganography method based on Markov chains. *Autom Control Comp S* 2016; 50(8): 802–808.

9. Go A, Bhayani R and Huang L. Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford 2009; 1(12): 2009.

10. Maas AL, Daly RE, Pham PT, et al. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, Portland, OR, 19–24 June 2011. Uppsala: Association for Computational Linguistics.

11. Thomson A. News dataset, 2005, https://www.kaggle.com/snapcrack/all-the-news/data

12. Murphy B and Vogel C. The syntax of concealment: reliable methods for plain text information hiding. In: Delp EJ III and Wong PW (eds) *Security, steganography, and watermarking of multimedia contents IX*, vol. 6505. Berlin: Springer, 2007, p.65050Y.

13. Stutsman R, Grothoff C, Atallah M, et al. Lost in just the translation. In: *Proceedings of the 2006 ACM symposium on applied computing*, Dijon, 23–27 April 2006, pp.338–345. New York: ACM.

14. Chen X, Sun H, Tobe Y, et al. Coverless information hiding method based on the Chinese mathematical expression. In: *International conference on cloud computing and security*, Nanjing, China, 13–15 August 2015, pp.133–143. Berlin: Springer.

15. Zhou Z, Mu Y, Zhao N, et al. Coverless information hiding method based on multi-keywords. In: *International conference on cloud computing and security*, Nanjing, China, 29–31 July 2016, pp.39–47. Berlin: Springer.