A Comprehensive Review on Deep Learning-based Generative Linguistic Steganography

Dr khaled nagaty
khaled.nagaty@bue.edu.eg

Israa Lotfy Lotfy
The British University in Egypt, israa.lotfy@bue.edu.eg

Abeer Hamdy Dr.
The British University in Egypt, Abeer.Hamdy@bue.edu.eg

A Comprehensive Review on Deep Learning-Based Generative Linguistic Steganography

Israa Lotfy Badawy(✉), Khaled Nagaty, and Abeer Hamdy

The British University in Egypt, Cairo, Egypt
{israa.lotfy,khaled.nagaty,abeer.hamdy}@bue.edu.eg

Abstract. The recent development of deep learning has made a significant breakthrough in linguistic generative steganography. The text has become one of the most intensely used communication carriers on the Internet, making steganography an efficient carrier for concealing secret messages. Text steganography has long been used to protect the privacy and confidentiality of data via public transmission. Steganography utilizes a carrier to embed the data to generate a secret unnoticed and less attractive message. Different techniques have been used to improve the security of the generated text and quality of the steganographic text, such as the Markov model, Recurrent Neural Network (RNN), Long short-term memory (LSTM), Transformers, Knowledge Graph, and Variational autoencoder (VAE). Those techniques enhance the steganographic text's language model and conditional probability distribution. This paper provides a comparative analysis to review the key contributions of generative linguistic steganographic deep learning-based methods through different perspectives such as text generation, encoding algorithm, and evaluation criteria.

Keywords: Text steganography · Information hiding · Deep learning

1 Introduction

With the increasingly deep innovation of technology and the Internet, securing information is becoming a complex issue against the domination of hackers and espionage. Various techniques have been proposed to protect the transmission of sensitive information via public and private communication channels; nevertheless, none of those techniques managed to prevent 100% of security threats. There are three essential methods of information security systems to address those challenges: cryptography, watermarking, and steganography [11]. Cryptography is an encryption technique that converts secret information into an enciphered form with the help of an encryption key. The third party can easily observe the existence of encrypted data that contains a transformation of the original information. The second method is Watermarking, which secures the originality of information by incorporating a signature in the original information. The signature cannot be directly detected without the proper techniques. In order to overcome the shortcomings of watermarking and cryptography, steganography approaches have been

M. E. Auer et al. (Eds.): ICL 2022, LNNS 633, pp. 651–660, 2023.
https://doi.org/10.1007/978-3-031-26876-2_61

proposed. Steganography is the art and science of concealing critical information in different covers, such as text, image, video, and audio, without any suspicion about the existence of the information [22]. Text is the most popular communication carrier in people's daily lives. 98.1% of users on social media utilize text messaging to communicate [13]. The text may include confidential information, so it needs information hiding techniques to hide it and make it available in a more secure way. Text Steganography is relatively the most complicated cover medium compared to other types because of the lack of redundant information in the text. To clarify, a slight modification in the text file causes the text file to be no longer meaningful or grammatically valid.

Text Steganography is a type of Hiding Information to conceal confidential data without being conscious of its existence. The output from the text steganography system can be called stego text or steganographic text [22]. It can be defined into three categories [11]: A) Text-modification-based Steganography. B) Text-selection-based Steganography. C) Text-generation-based Steganography. The text-selection-based steganography mainly embeds the secret message by constructing a text corpus used to hide the secret message. This corpus selects suitable steganographic text based on keywords and labels. The hiding process in the text-modification approach concentrates on manipulating text characteristics or content through synonym substitution, changing the line spacing, format redundancy, and syntactic change. The text-generation-based steganography secures the hidden message by generating a new cover text using NLP techniques. So it does not require cover in advance. To review, the steganography technique aims at hiding the information without any suspicion of the transmission of a secret message. It is apparent that if any suspicion is raised, so the goal of this approach is defeated. As a result, steganography utilizes a cover or carrier to embed the data to make the secret message unnoticed and less attractive.

With the significant development of deep learning and natural language processing (NLP) techniques, researchers have successively proposed different approaches to achieve a larger hiding capacity for generative text steganography. However, the quality of the generated long text is still poor. Nevertheless, text steganography enhances the hiding capacity by manipulating language characteristics, grammatical or orthographic. This study reviews the evolution of text steganography by investigating and exploring research papers related to automatic generative text steganography. The contributions of this research are defined as follows:

– It provides a comparative analysis of the existing linguistic text steganography deep learning-based approaches based on different perspectives such as text generation, encoding algorithm, and evaluation criteria.
– It discusses the future directions in this research field.

The rest of this paper is structured as follows. Section 2 discusses the deep learning methodology used to generate a steganography system, focusing on the encoding process and evaluation metrics. Section 3 provides future work recommendations and the findings from the existing methods, and Sect. 4 briefs the evaluation criteria for generative linguistic steganography. Section 5 concludes this paper.

2 Deep Learning-Based Linguistic Steganography

Different techniques have been proposed to secure confidential information without suspicion about the message's existence. Markov model is one of the earliest techniques developed in this area. Yang et al. [14] designed a linguistic steganographic system based on Markov Model and Huffman Coding to generate text steganography automatically. The generated system is divided into two stages: the automatic generation of the text and the embedding process for the secret message. The generated text should follow the statistical language distribution of the training sample, so the Markov chain model is used to ensure the correlation between the generated text and the statistical distribution. At the same time, the construction of the Huffman tree is implanted for each iteration according to the different conditional probability distributions of each word to complete the embedding process. This approach helps to ensure the information hiding, but the results of the embedding rates are insufficient as while the embedding rate increases, the statistical distribution difference between the training data and generated text gradually increases. The proposed model was evaluated on three Twitter, movie reviews, and news datasets. The model's accuracy while tested on the Twitter dataset is 56% for embedding rate (4bit/word). However, the deep learning and NLP techniques achieve a definite improvement, illustrated in the paragraphs below.

Yang et al. [10] hypothesized that using the recurrent neural networks in linguistic steganography conduct high-quality text covers. At the same time, the generated text is more natural and of higher quality. By learning the statistical language distribution model, this paper automatically generates the stego text based on RNN. This distribution learns from a massive volume of standard text to produce messages that follow the statistical patterns. Embedding the information depends on the conditional probability distribution of each word coded using the binary tree technique and Huffman tree during the synthesis sentence. This approach increased the embedding rate and decreased the system security since it analyses the statistical features of a single phrase, abandoning the broader distribution of batch-generated texts. The proposed system's performance is measured through three datasets: Twitter, news, and movie reviews. The model shows promising results while the embedding rate increases 7bits/word, equaling 17.13%. The system's accuracy against the steganalysis technique proposed in [7] is 52% for 4 bit/word.

Yang et al. [9] proposed different strategies to improve the generation of semantically controllable steganographic text and the encoder-decoder framework. This work has evaluated three different encoding techniques Gated Recurrent Unit (GRU) model, the Transformer model, and the Topic-Aware model. The candidate pool (CP) used for text generation has been generated by categorical sampling to achieve the goal of generation semantic text. The authors used METEOR and ROUGH-L matric to evaluate the semantics of the generated text (the higher, the better). The experiment analysis for this system used ROC stories datasets.

Xiang et al. [4] demonstrated that embedding the secret message based on the character level will improve the hiding capacity and message security. So the authors propose a linguistic steganography system based on the LSTM-based character-level language model; this model will provide the prediction of the following character instead of the next word. The information hiding process will go through two dependent directions.

The first process is to generate the cover text based on the secret message to generate multiple cover texts. The second direction is selecting stego text from all candidates with high quality. The selection process is defined regarding the perplexity calculation for all candidates. The experimental results of this work proved that the proposed system achieves a faster running speed as it takes 0.642 s to generate the stego text compared to RNN- Stega [10], which takes 3.25 s. A more significant embedding rate has been achieved with 12.56% capacity.

Based on the ideas of Kang et al. [5], combining the LSTM network and attention mechanism with keywords improved the stego text quality. The proposed combination uses a large-scale text database to generate a language model. In-text generating process, the prediction of the next word is defined according to the conditional probability distribution; this calculation is performed by the LSTM network and the secret value to be embedded. Keywords are considered with the attention mechanism technique to improve the quality of the generated text. Moreover, they point out that the steganographic text's quality mainly depends on the training dataset.

Furthermore, the semantic quality will be poor if the dataset is on a small scale. This paper showed that the difference between LSTM and LSTM with attention mechanism is slightly different as the time needed to generate the steganographic text is averagely the same. The time needed for the VLC method (Huffman Tree) is more than that of the FLC method (Perfect Binary Tree). The reason is that the method of VLC spends more time generating the Huffman tree. Nevertheless, FLC takes around 3.7 s to generate the text (100-word) to build the binary tree, which presents a good performance in data embedding efficiency.

Probability-based adaptive embedding algorithm is responsible for defining the candidate word space and the embedding capacity based on the similarity of word probability. This approach focuses on the most considerable transition probability to embed words. Zhou et al. [2] assert that using the adaptive probability distribution with the generative adversarial network (GAN) achieves high-security performance and high quality. In addition, the adaptive embedding algorithm with a similarity function can keep the embedded distribution consistent with the accurate distribution. Moreover, embedding information is considered in the training process instead of the isolation between the generation stages.

Furthermore, the generated information hiding model reduces the embedding deviation and improves performance. The system was evaluated against the steganalysis technique proposed in [19] with an accuracy of 66% for hiding 3 bit/wors. The author used three datasets for the training side: Twitter, Microsoft Coco, and Movie Reviews.

Yang et al. [1] follow the encode-decoder architecture to perform the new linguistic steganography system using a variational auto encoder (VAE-Stega). The proposed system's encoder learns the normal text's overall statistical distribution, and the decoder generates steganographic text. A large-scale database (Twitter and movie reviews) has been used to confirm the proposed system's performance. This work compares two different encoders, LSTM and Bidirectional Encoder Representations from Transformers (BERT). All results of this system compared to RNN-Stega [10] and RNN [9] show which one is more secure. The paper showed the results of the steganalysis technique in [17] over the proposed method using arithmetic coding (AC) and Huffman tree (HT) and showed that AC could persist with an accuracy of 62% compared to HT accuracy of 62%. The finding from this paper is that the arithmetic coding can perform slightly better than the Huffman tree.

Ziegler et al. [8] combined the pre-trained language model, which is called generative pre-trained transformer (GPT), with the arithmetic coding (AC) to develop the steganographic system. The improvement of AC controls the difference in the conditional probability distribution between normal text and steganographic texts.

Shen et al. [12] proposed a new steganography system that encodes the secret message based on arithmetic coding with the help of a pre-trained language model. The proposed method improves the imperceptibility of the secret message compared to previous methods. This paper does not use the normal arithmetic coding as it may generate a rarely-used cover text token; it proposes a new self-adjusting arithmetic coding (SAAC) to overcome this issue. This work was evaluated using Drugs, News, COVID-19, and Random datasets. This work achieved better results than RNN-Stega [10], but the generated text contains some factual errors.

Li et al. [6] point out the problem of semantics in text steganography. Hence, they propose a linguistic system based on the knowledge graph to generate a steganographic text on a specific topic by encoding the entitles and relationships data. The proposed solution goes along with the transformer architecture (encoder and decoder). The graph encoding built the graph vectors based on the topic and content at the encoder process. Then those vectors are used to generate steganographic text at the decoder process. The system was evaluated using the steganalysis approach proposed in [17] with an accuracy of 67%. The system was evaluated using METEOR matric, which achieves significant results.

3 Discussion and Research Direction

The steganographic systems aim to hide the secret information in a carrier without any suspicion about the existence of the information and then securely extract the information from the carrier. Various models have been proposed to improve the area of generative linguistic text steganography. Some models improve the system's performance by taking less time while generating the carrier message. Others achieved good results in embedding capacity and the quality of the generated sentences. However, improving the quality of generated sentences does not mean that the steganographic text is secure. With the developments of steganalysis, the proposed steganographic models cannot survive.

Furthermore, the semantic expression of the steganographic text needs to be controlled to generate controllable semantics text in a specific context. The proposed frameworks of linguistic steganography face fundamental challenges. The first challenge, while the embedding rates increase, the quality of the generated text decreases. Besides, meaningless sentences with grammatical errors appear. For that reason, building a language generation model and generating well-quality text carriers with smoother sentences is a problem that needs to be addressed. The second challenge is that the generated text from the above frameworks is only generated based on the statistical distribution of probability. The semantics, emotions, and topics of the generated text are uncontrollable.

Consequently, the semantics, emotions, and topics of the generated text should be considered to improve the quality of new text generation and enhance sentence fluency. We point out that encoder-decoder architecture can positively influence the hiding process of text steganography. The encoder concentrates on learning the statistical distribution of the normal texts, and the decoder generates the sentences based on the outcomes from the encoder. The most effective encoding techniques in steganography are Huffman Tree, Binary Tree, and Arithmetic coding. The binary tree takes less time to produce stego text than the Huffman tree while only constructing a Perfect binary tree but the steganographic text generated by the Huffman tree is better than the binary tree. Future directions to explore the language models and encoder-decoder architecture may improve the current gaps in the steganography area.

Nevertheless, Arithmetic encoding achieved significant results in the concealment of the information. The authors used different datasets to evaluate and construct the steganographic model. Twitter, News, and Movie Reviews datasets are the most used for training. Table 1 illustrates each paper's encoding and deep learning techniques. The future work in this area can be defined as follows:

- Improve the quality of steganographic text by controlling the text semantics to generate a controllable text generation.
- Reimburse attention to the encoder-decoder architecture to enhance the security of the text.
- Optimize the encoding techniques to minimize the difference in the probability distribution.

4 Evaluation Criteria

The purpose of text steganography is to hide the existence of information in the carrier to ensure the concealment of important information. The previous works analyzed the performance of their systems in three different aspects: information hiding efficiency, hidden capacity, and information imperceptibility. For Semantic expressions, two methods can be applied bilingual evaluation understudy (BLEU), recall-oriented understanding for gisting evaluation-longest common subsequence (ROUGH-L), and metrics for evaluation of translation with explicit ordering (METEOR) [6].

4.1 Information Hiding Capacity

Information Hiding Capacity measures how long the model takes to hide the secret information. Different aspects affect the information hiding capacity, such as the dictionary size, candidate pool, and the encoding process algorithm.

Paper [13] mentioned a comparison between the different encoding techniques with different candidate pools and found that the Perfect binary tree takes less time when generating 50 words with candidate pool 332 with an average of 4.854 s. The perfect binary tree shows a better result than the Huffman tree regarding information hiding capacity.

Table 1. Previous work in generative linguistic text steganography

Technique	Year	Author	Encoding process	Dataset	Steganalysis
RNN	2019	Yang et al. [10]	Huffman Tree & Perfect Binary Tree	Twitter, Movie Reviews, News	[7]
	2021	Yang et al. [9]	(GRU) & Transformer model & Topic-Aware model	ROC Stories	[7, 15, 16]
RNN + Knowledge Graph	2021	Li et al. [6]	Graph Encoding	AGENDA	[17]
LSTM	2020	Xiang et al. [4]	LSTM Character-level Language Model	Gutenberg corpus	-
LSTM + Attention Mechanism	2020	Kang et al. [5]	Huffman Tree & Perfect Binary Tree	Zhihu, ESSAY	[7]
LSTM + GAN	2021	Zhou et al. [2]	Huffman Tree & Perfect Binary Tree	Twitter, Microsoft Coco, and Movie Reviews	[19–21]
LSTM + Language Models (GPT & BERT)	2019	Ziegler et al. [8]	Arithmetic Coding	CNN/Dailymail (CNNDM)	-
	2020	Shen et al. [12]	Self Adjusting Arithmetic Coding	Drug, News, COVID-19, Random	-
Variational autoencoder (VAE)	2021	Yang et al. [1]	Huffman Tree Arithmetic Coding	Twitter Movie review	[17–19]

4.2 Hidden Capacity Analysis

Hidden capacity analysis refers to the embedding rate (ER), which calculates how much information can be embedded in texts. The embedding rate measurement divides the number of embedded bits by the number of bits occupied by the generated text. In other words, it calculates the average number of bits concealed in each word (Bwp – Bits/word) [14]. The result of ER constructs an opposite relationship with the hiding process. The formalization of the embedding rate is as follows (1):

$$\mathrm{ER} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\frac{K_i}{L_i} \tag{1}$$

where N is the number of the generated sentences, K_i is the number of the bits embedded in i-th sentences, and L_i is the length of the i-th sentences.

4.3 Information Imperceptibility

Information imperceptibility is an essential aspect of the evaluation process to ensure the concealing of confidential information [23]. This evaluation will evaluate the quality of the generated text, text statistical distribution characteristics, and anti-steganalysis ability.

Perplexity Metric (ppl)

Measures the generated sentences' quality and the language model of the generated sentences. The smaller its value is, the better the generated text's quality is with the training data's statistical distribution. Perplexity is the primary measure of steganographic text in NLP.

$$\text{perplexity} = 2^{-\frac{1}{n} log\, p(s_i)} \tag{2}$$

$$= 2^{-\frac{1}{n}\log p(w_1, w_2, w_{3,\ldots,} w_n}$$

$$2^{-\frac{1}{n}\sum_{j=1}^{n}\log p(w_i|w_1, w_{2,\ldots.,} w_{j-1}}$$

where $si = \{w1,w2,w3,\ldots,wn\}$is the generated sentence, $p(si)$ indicates the probability distribution, and n is the length of the generated sentences. As mentioned by Yang et al. [200], the difference in the statistical distribution between the generated steganographic text and the training texts can be evaluated using Kullback-Leibler divergence (KLD), Jensen-Shannon divergence (JSD), and Wasserstein Distance known as Earth Mover's Distance (EMD). KLD and JSD calculate the overall distribution of the generated and normal sentences in terms of statistical distribution to measure the security of the generated steganographic model.

Steganalysis Ability

This process aims to evaluate the performance of the generated steganographic sentences to resist steganalysis. Different factors, such as Accuracy (Acc), Precision (P), and Recall (R), can be used to calculate the resistance.

Accuracy. Calculates the proportion of both true positives and true negatives:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{3}$$

Precision. Measures the proportion of positive cases in the classified samples:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Recall. Measures the proportion of positives that are correctly identified as such:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

TP (True Positive) refers to the number of positive samples predicted as positive samples, and FP (False Positive) defines the number of negative samples predicted as negative samples. FN (False Negative) indicates the number of positive samples predicted to be negative, and TN (True Negative) represents the number of negative samples predicted to be negative.

5 Conclusion

Automatic generative linguistic steganography is a challenging and promising research topic in information security. Generative linguistic steganography aims to generate a cover text based on a secret message close to the humans' normal text. Text Steganography is relatively more complicated than other steganography types; because of the lack of redundant information in a text file compared with other types. Despite the immense improvements in this field in recent years, there remains a massive space for developing and enhancing this domain. As a comprehensive review of the deep learning-based approach in text steganography, this paper focus on the existing approaches, the encoding process, the evaluation techniques, and the improvements.

References

1. Yang, Z.-L., Zhang, S.-Y., Hu, Y.-T., Hu, Z.-W., Huang, Y.-F.: VAE-Stega: linguistic steganography based on variational auto-encoder. IEEE Trans. Inf. Forensics Secur. **16**, 880–895 (2021). https://doi.org/10.1109/TIFS.2020.3023279
2. Zhou, X., Peng, W., Yang, B., Wen, J., Xue, Y., Zhong, P.: Linguistic steganography based on adaptive probability distribution. IEEE Trans. Dependable Secure Comput. **19**, 2982–2997 (2021)
3. Alanazi, N., Khan, E., Gutub, A.: Efficient security and capacity techniques for Arabic text steganography via engaging Unicode standard encoding. Multimedia Tools Appl. **80**(1), 1403–1431 (2020). https://doi.org/10.1007/s11042-020-09667-y
4. Xiang, L., Yang, S., Liu, Y., Li, Q., Zhu, C.: Novel Linguistic steganography based on character-level text generation. Mathematics **8**(9), 1558 (2020). https://doi.org/10.3390/math8091558
5. Kang, H., Wu, H., Zhang, X.: Generative text steganography based on LSTM network and attention mechanism with keywords. Electron. Imaging **2020**(4), 291-1–291-8 (2020)
6. Li, Y., Zhang, J., Yang, Z., Zhang, R.: Topic-aware neural linguistic steganography based on knowledge graphs. ACM/IMS Trans. Data Sci. **2**(2), 1–13 (2021)
7. Samanta, S., Dutta, S., Sanyal, G.: A real time text steganalysis by using statistical method. In: Proceedings IEEE International Conference on Engineering and Emerging Technologies (ICETECH), pp. 264–268 (2016)
8. Ziegler, Z.M., Deng, Y., Rush, A.M.: Neural linguistic steganography. arXiv:1909.01496 (2019). http://arxiv.org/abs/1909.01496
9. Yang, Z., Xiang, L., Zhang, S., Sun, X., Huang, Y.: Linguistic generative steganography with enhanced cognitive-imperceptibility. IEEE Signal Process. Lett. **28**, 409–413 (2021). https://doi.org/10.1109/LSP.2021.3058889
10. Yang, Z., Guo, X., Chen, Z., Huang, Y., Zhang, Y.: RNN-Stega: linguistic steganography based on recurrent neural networks. IEEE Trans. Inf. Forensics Secur. **14**(5), 1280–1295 (2019)

11. Li, F., Tang, H., Zou, Y., Huang, Y., Feng, Y., Peng, L.: Research on information security in text emotional steganography based on machine learning. Enterp. Inf. Syst. **15**(7), 984–1001 (2020)
12. Shen, J., Ji, H., Han, J.: Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding. In: EMNLP (2020)
13. Gurunath, R., Alahmadi, A., Samanta, D., Khan, M., Alahmadi, A.: A novel approach for linguistic steganography evaluation based on artificial neural networks. IEEE Access **9**, 120869–120879 (2021)
14. Yang, Z., Jin, S., Huang, Y., Zhang, Y., Li, H.: Automatically generate steganographic text based on Markov model and Huffman coding, November 2018. arXiv:1811.04720. http://arxiv.org/abs/1811.04720
15. Meng, P., Hang, L., Yang, W., Chen, Z., Zheng, H.: Linguistic steganography detection algorithm using statistical language model. Proc. Int. Conf. Inf. Technol. Comput. Sci., 540–543 (2009)
16. Chen, Z., et al.: Linguistic steganography detection using statistical characteristics of correlations between words. In: Proceedings International Workshop on Information Hiding, pp. 224–235 (2008)
17. Yang, Z., Huang, Y., Zhang, Y.-J.: A fast and efficient text steganalysis method. IEEE Signal Process. Lett. **26**(4), 627–631 (2019)
18. Din, R., et al.: Performance analysis on text steganalysis method using a computational intelligence approach. In: Proceedings International Conference of Electrical Engineering, Computer Science and Informatics (EECSI), pp. 19–20 (2015)
19. Wen, J., Zhou, X., Zhong, P., Xue, Y.: Convolutional neural network based text steganalysis. IEEE Signal Process. Lett. **26**(3), 460–464 (2019)
20. Niu, Y., Wen, J., Zhong, P., Xue, Y.: A hybrid R-BILSTM-C neural network based text steganalysis. IEEE Signal Proc. Lett. **26**(12), 1907–1911 (2019)
21. Yang, Z., Wang, K., Li, J., Huang, Y.: TS-RNN: text steganalysis based on recurrent neural networks. IEEE Signal Proc. Lett. **26**(12), 1743–1747 (2019)
22. Thabit, R., Udzir, N., Yasin, S., Asmawi, A., Roslan, N., Din, R.: A comparative analysis of Arabic text steganography. Appl. Sci. **11**(15), 6851 (2021)
23. Xiang, L., Wang, R., Yang, Z., Liu, Y.: Generative linguistic steganography: a comprehensive review. KSII Trans. Internet Inf. Syst. **16**(3), 986–1005 (2022). https://doi.org/10.3837/tiis.2022.03.013