Journal of Cyber Security
DOI:10.32604/jcs.2020.015010
Article

Tech Science Press

A Two-Stage Highly Robust Text Steganalysis Model

Enlu Li[1], Zhangjie Fu[1,2,3,*], Siyu Chen[1] and Junfu Chen[1]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[2]Guangxi Key Laboratory of Cryptography and Information Security, Guilin, 541004, China
[3]College of Information Science and Technology, College of Cyber Security, Jinan University, Guangzhou, 510632, China
[*]Corresponding Author: Zhangjie Fu. Email: fzj@nuist.edu.cn

Received: 02 November 2020; Accepted: 10 November 2020

Abstract: With the development of natural language processing, deep learning, and other technologies, text steganography is rapidly developing. However, adversarial attack methods have emerged that gives text steganography the ability to actively spoof steganalysis. If terrorists use the text steganography method to spread terrorist messages, it will greatly disturb social stability. Steganalysis methods, especially those for resisting adversarial attacks, need to be further improved. In this paper, we propose a two-stage highly robust model for text steganalysis. The proposed method analyzes and extracts anomalous features at both intra-sentential and inter-sentential levels. In the first phase, every sentence is first transformed into word vectors. To obtain a high dimensional sentence vector, we use Bi-LSTM to obtain feature information for all words in the sentence while retaining strong correlations. In the second phase, we input multiple sentences vectors into the GNN, from which we extract inter-sentential anomaly features and make a judgment as to whether the text contains secret messages. In addition, to improve the robustness of the model, we add adversarial examples to the training set to improve the robustness and generalization of the steganalysis model. Theoretically, our proposed method is more robust and more accurate in detection compared to existing methods.

Keywords: Text steganalysis; adversarial attack; natural language processing; deep learning

1 Introduction

Steganography is a technique that embeds secret messages into carriers for covert communication. In contrast to steganography, steganalysis is used to detect the presence of secret messages in a carrier. These two technologies have always developed together in competition. The steganography methods try to preserve the characteristics of the carrier by making undetectable changes so that no one other than the recipient can perceive the secret messages. Media used for steganography include images, text, audio, video, etc. With the development of social networking platforms, textual information such as e-books, online news, news reviews, and user data has increased dramatically and prompts the development of research related to textual implicit writing.

From the perspective of the carrier, text steganography methods can be divided into two main classes, modification-based and synthesis-based. Modification-based steganography methods implement the embedding of secret messages by modifying the original text. Specific steganography methods include synonym substitution [1–2], syntactic transformation which means changing the sentence structure according to syntax rules [3], changing the expression according to the original semantic knowledge [4–5], etc. However, it is difficult for such methods to maintain a balance between generating a reasonable and appropriate stego text and a high embedding capacity. With the rapid development of deep learning in the field of natural language processing, it is simple to generate large amounts of high-quality text. Researchers use these techniques to synthesize text for steganography [6–10]. Such methods take advantage of the powerful feature extraction capabilities of neural networks to learn the statistical distribution features of text. During the process of text steganography, generate text containing secret information that is consistent with the statistical features of the training text.

The development of social networks has provided a rich medium for text steganalysis. However, if terrorists use text steganography to transmit terrorist messages, it would greatly endanger the security of society. Therefore, it is necessary to study text steganalysis methods. Traditional text steganalysis methods [11–15] usually extract manual features based on domain knowledge and determine whether the text contains secret messages based on whether the features are anomalous or not. These methods rely on human experience and the features extracted are limited. As a result, it is difficult to adapt to progressively complex text steganography methods. Current deep learning-based methods for text steganalysis [16–23] achieve better performance than hand-crafted text steganalysis methods by automatically extracting and learning high-dimensional features.

Neural networks have shown great potential in the field of steganography. But there are still many challenges for the text steganalysis method based on neural network. Most text steganalysis methods only analyze whether the text contains secret messages from the perspective of sentences, ignoring the inter-sentential correlation features. The cover synthesis-based text steganography methods may not exhibit unusual features within sentences, but rather exhibit semantic anomalies between sentences. At the same time, with the application of adversarial examples in steganography [24,25], steganography methods already have the ability to actively deceive steganalysis models. Text steganalysis methods need to be further improved to be resistant to adversarial attacks. In this paper, we propose a two-stage highly robust text steganalysis model based on long short-term memory (LSTM) [26] and graph neural network (GNN) [27]. The proposed method analyzes and extracts anomalous features at both intra-sentential and inter-sentential levels. We generate corresponding adversarial examples based on existing neural network-based text steganography methods. The generated adversarial examples are added to the training set to improve the robustness and generalization of the steganalysis model.

2 Related Work

2.1 Text Steganalysis

In recent years, text hiding in the form of watermarking and steganography has been widely used in convert communication, copyright protection, content authentication, and other fields. Relative to text steganography, text steganalysis is the process and science of identifying whether there is hidden information in a given carrier text. For example, Yang et al. [19] proposed a fast and efficient method for the automatic generation of stego text by neural networks. The correlation between words is mapped to semantic space, and then the correlation between words in text is analyzed and extracted with the hidden layer. Niu et al. [20] introduced the one-dimensional asymmetric convolution and residual module on the basis of Bi-LSTM [28], and realized the discrimination of inter-sentential relations while extracting the semantic information within the sentence. Shortcut block is introduced to alleviate the loss of feature relation between sliding windows caused by convolution operation to some extent, so that the model effect is improved slightly. Xiang et al. [21] proposed a steganographic analysis model based on two-stage CNN for the problems of synonym substitution in steganographic analysis, which is composed of sentence-level CNN and text-level CNN, respectively corresponding to the sentence-level detection and full-text detection of the article to be tested. This kind of step method is not common in steganographic analysis, which has its own characteristics and has significance for further study. Therefore, this paper carries out study by referring to this method.

2.2 Adversarial Examples

There has been a great deal of research on adversarial examples in the field of machine vision [29–32]. In natural language processing, the research of adversarial examples is also very important. It achieves the purpose of spoofing neural networks by making small changes to the original sample. Designing training methods that resist adversarial examples can improve the robustness and generalization of the model.

In 2013, Szegedy et al. [29] first proposed an algorithm named BFGS to generate adversarial examples in the field of deep learning. The paper pointed out that the cause of adversarial example may be the nonlinear expression and overfitting of the model. In 2014, Goodfellow et al. [30] argued that the linear characteristics of deep neural networks in high-dimensional space are the fundamental cause of adversarial example formation and designed a fast and efficient method FGSM for generating adversarial samples based on this theory. For a given network *N*, the input text *C*, θ represents the paraments of the network, *y* means the target label. We use $L_N(\theta, C, y)$ indicates the loss of the network, η represents the gradient of L_N:

$$\eta = \nabla_C L_N(\theta, C, y) \tag{1}$$

It can be obtained by backpropagation. Multiply η by a coefficient to get the perturbation.

In text steganography, the same method is used to find the word with high impact of the most loss function. Sorting these words and then replacing, deleting, or inserting them in order, the input can be transformed into adversarial examples to trick the steganalysis. Therefore, text steganalysis methods need to be further improved to be resistant to adversarial attacks.

2.3 Graph Neural Networks

Deep learning is good at working with structured data such as audio, images, and text. However, not everything can be represented as sequences or grids, such as knowledge graphs, social network relationships, and so on. This prompted the emergence and development of graph neural networks.

In 2017, Kipf et al. [27] proposed the first graph neural networks, which applies convolution in image processing to graph structures for the first time. The structure is shown in Fig. 1. In each layer of the neural network, each node combines the features of neighbor nodes and then does a linear transformation. Stacking of multiple network layers followed by node classification, prediction, and other tasks.

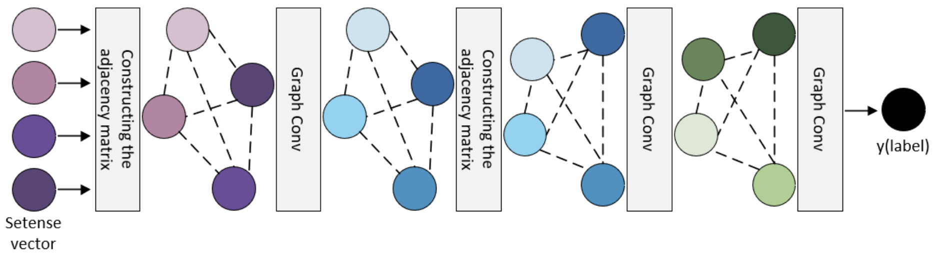
Setense vector
Constructing the adjacency matrix
Graph Conv
Constructing the adjacency matrix
Graph Conv
Graph Conv
y(label)

Figure 1: The structure of graph neural networks

However, GNN needs to know information about the structure of the entire graph, including the predicted nodes, which is not consistent with the real task. In addition, GNN also has problems such as memory consumption. To solve these problems, more graph neural networks are proposed [33–35].

3. The Proposed Method

3.1 The Overall Architecture

The two-stage highly robust text steganalysis model analyzes and extracts anomalous features at both intra-sentential and inter-sentential levels. In the first phase, every sentence $[s_1, s_2, \cdots s_m]$ in the text is first transformed into word vectors $[x_1, x_2, \cdots x_n]$ using word embedding. To obtain a high dimensional sentence vector $[S_1, S_2, \cdots S_m]$, we used Bi-LSTM to obtain feature information for all words in the sentence while retaining strong correlations. In the second phase, we input multiple sentence vectors into the GNN, from which we extract inter-sentential anomaly features and make a judgment as to whether the text contains secret messages. To improve the robustness of the model, we generate corresponding adversarial examples based on existing neural network-based text steganography methods. The generated adversarial examples are added to the training set to improve the robustness and generalization of the steganalysis model. The framework of the two-stage highly robust text steganalysis model is illustrated in Fig. 2.

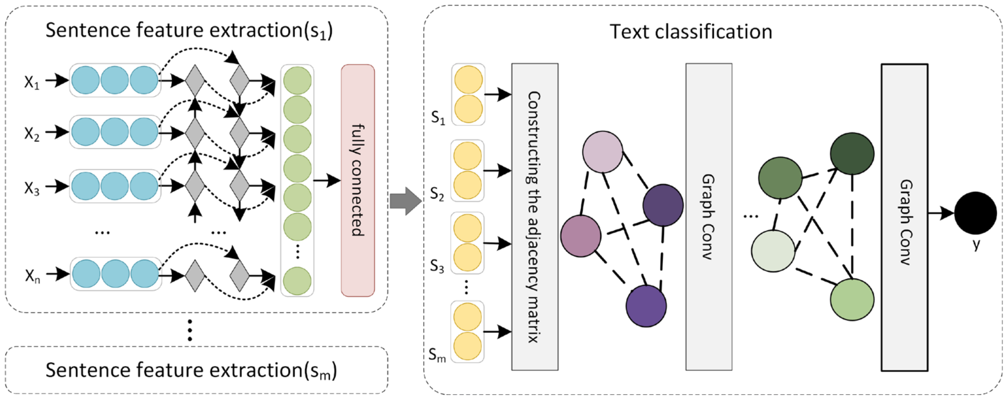
Sentence feature extraction(s1)
X1
X2
X3
Xn
fully connected
Sentence feature extraction(sm)
Text classification
S1
S2
S3
Sm
Constructing the adjacency matrix
Graph Conv
Graph Conv
y

Figure 2: The framework of the proposed network

3.2 Sentence Feature Extraction

Text is first transformed into word vectors using word embedding. Text consists of m sentences, we denote a text as $[s_1, s_2, \cdots s_m]$. To improve the robustness of the model, there are adversarial examples in these sentences. Each sentence consists of n words and we denote each sentence as $[x_1, x_2, \cdots x_n]$. Since the network cannot process text directly, we transform the words and phrases in each sentence into a low-dimensional continuous vector. In the field of natural language processing, such a technique is known as word embedding.

Features of multiple word vectors are extracted and transformed into feature vectors of sentences using Bi-LSTM. Similar to RNN [36], LSTM makes use of context-sensitive information in the mapping process between input and output sequences. It also compensates for the shortcoming of RNN: the limited scope of accessing contextual information. However, LSTM cannot encode back-to-front information, which is not conducive to more fine-grained classification. Therefore we use Bi-LSTM to extract the features of the sentence as a whole. The extracted sentence feature vector can be represented as $[S_1, S_2, \cdots S_m]$. The extraction process can be expressed as:

$$\begin{aligned} S_i &= f(\theta, s_i) \\ &= f(\theta, [x_1, x_2, \cdots x_n]) \end{aligned} \tag{2}$$

where s_i represents the i-th sentence, which can be represented by the vector $[x_1, x_2, \cdots x_n]$. $f(\bullet)$ represents the Bi-LSTM network and θ means that all trainable parameters in the network. S_i is the output of the network, i.e., the feature vector of the i-th sentence.

3.3 Text Classification

Most text steganalysis models use CNN to further purify the features and classify the text after obtaining the feature vectors of the sentences. These approaches focus only on anomalous features within sentences and rarely consider semantic correlations between sentences in terms of the text as a whole. We input multiple sentences vectors $[S_1, S_2, \cdots S_m]$ into the GNN, from which we extract inter-sentential anomaly features and make judgments. Each node in the graph represents a sentence vector, and the edges represent the sentence-to-sentence correlations. This is a graph-level task that does not only depend on the properties of nodes or edges. Each change in graph space features incorporates correlations between neighboring nodes. The classification of text is achieved from the overall structure of the graph.

3.4 Training Framework

During the training process, we update the network parameters using backpropagation algorithm. The cross entropy error loss is used as the loss function of the network, which can be described as:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\left[p_i \log(q_i) + (1-p_i)\log(1-q_i)\right] \tag{3}$$

where N is the batch size of the texts. p_i represents the ground truth label of the text. q_i is the predict label of the text. i stands for the i-th sample in each batch. Thus, the entire training is a supervised learning process. The network is brought to an optimal state by minimizing the loss function.

4 Experiments Analysis

4.1 Experimental Settings

The experiments were conducted on Gutenberg [37]. T-Lex is a typical text steganalysis algorithm. We select it to generate the stego text. 8000 pairs of cover text and sego text as training set, 2000 pairs of cover text and sego text as test set. In the training phase, we chose Adam as the optimization method, cross entropy loss is used as the loss function. The learning rates were set as 0.001, the batch size was set as 64. It is worth noting that in order to improve the robustness of the text steganography algorithm, we randomly replace some stego text in the training set using the corresponding adversarial examples. The experiments were conducted on an NVIDIA GeForce RTX 2080Ti GPU card which has 11GB memory.

4.2 Experimental Analysis

In order to evaluate our proposed two-stage highly robust steganalysis model, we use metrics commonly used in steganalysis to measure the performance of the model. The three evaluation indicators include Precision, Recall, Accuracy. All original text is denoted by cover, text containing secret messages is denoted by stego.

4.2.1 Precision

Precision denotes the ratio of samples that were correctly classified as stego to samples classified as stego. The definition is formularized as follows:

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

where TP (True Positive) represents the number of correctly predicted stego sentences. FP (False Positive) means the number of cover sentences incorrectly predicted as stego sentences. Since we used graph neural networks to learn the characteristics of correlations between sentences, the number that cover was erroneously predicted to stego would be smaller, i.e., the FP would be smaller. Therefore, our proposed method will have a higher Precision-value compared to other methods.

4.2.2 Recall

Recall represents the ratio of all samples correctly classified as stego to all correctly classified samples. The definition is formularized as follow:

$$Recall = \frac{TP}{TP+TN} \tag{5}$$

where TN (True Negative) illustrates the number of correctly predicted cover sentences. The application of graph neural networks to text steganalysis combines the advantage that graph convolution can aggregate feature information from nearest neighbor nodes. It is therefore theoretically possible to make a more accurate determination on whether a text contains secret messages or not. As a result, TP and TN would be more accurate, and Recall would theoretically improve slightly.

4.2.3 Accuracy

Accuracy represents the ratio of all correctly classified samples to the total sample. The definition is formularized as follow:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where FN (False Negative) indicates the number of stego sentences incorrectly predicted as cover sentences. We used a poisoning attack when training our text steganalysis model. We added adversarial examples to the training data to deal with the possibility of adversarial attacks in real situations. Thus, the number of stego sentences incorrectly predicted as cover sentences would be smaller, i.e., the FN would be smaller. In summary, our method will theoretically have higher accuracy. We will further refine our experiment.

5 Conclusion

In this paper, we propose a two-stage highly robust text steganalysis model based on LSTM and GNN. The proposed method analyzes and extracts anomalous features at both intra-sentential and inter-sentential levels. Text is first transformed into word vectors using word embedding, and features of multiple word vectors are extracted and transformed into feature vectors of sentences using Bi-LSTM. We input multiple sentence vectors into the GNN, from which we extract inter-sentential anomaly features and make judgments. To improve the robustness of the model, we generate corresponding adversarial examples based on existing neural network-based text steganography methods. The generated adversarial examples are added to the training set to improve the robustness and generalization of the steganalysis model.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant U1836110, U1836208; by the Jiangsu Basic Research Programs-Natural Science Foundation under grant numbers BK20200039; by China Postdoctoral Science Foundation (2017M610574); by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China; by the Opening Project of Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201713), Guangdong Provincial Key Laboratory of Data Security and Privacy Protection (Grant No. 2017B03031004).

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[2] U. Topkara, M. Topkara and M. J. Atallah, "The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions," in *Proc. MM&Sec*, New York, NY, USA, pp. 164–174, 2006.

[3] L. Huo and Y. Xiao, "Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations," in *Proc. ICCC*, Chengdu, SC, China, pp. 2776–2780, 2016.

[4] C. Y. Chang and S. Clark, "Linguistic steganography using automatically generated paraphrases," in *Proc. NAACL HLT*, Los Angeles, CA, USA, pp. 591–599, 2010.

[5] M. Niimi, S. Minewaki, H. Noda and E. Kawaguchi, "A framework of text-based steganography using sd-form semantics model," in *Proc. IWDW*, Berlin, Germany, pp. 3–4, 2003.

[6] T. Fang, M. Jaggi and K. Argyraki, "Generating steganographic text with lstms," arXiv preprint arXiv: 1705. 10742, 2017.

[7] Y. Luo and Y. Huang, "Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry," in *Proc. MM&Sec*, Philadelphia, PA, USA, pp. 99–104, 2017.

[8] Z. Yang, P. Zhang, M. Jiang, Y. Huang and Y. J. Zhang, "Rits: Real-time interactive text steganography based

on automatic dialogue model," in *Proc. ICCCS*, Haikou, China, pp. 253–264, 2018.

[9] Z. L. Yang, X. Q. Guo, Z. M. Chen, Y. F. Huang and Y. J. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1280–1295, 2018.

[10] Z. M. Ziegler, Y. Deng and A. M. Rush, "Neural linguistic steganography," arXiv preprint arXiv: 1909. 01496, 2019.

[11] C. M. Taskiran, U. Topkara, M. Topkara and E. J. Delp, "Attacks on lexical natural language steganography systems," in *Proc. SPIE*, San Jose, CA, USA, 2006.

[12] H. Yang and X. Cao, "Linguistic steganalysis based on meta features and immune mechanism," *Chinese Journal of Electronics*, vol. 19, no. 4, pp. 661–666, 2010.

[13] Z. Chen, L. Huang, P. Meng, W. Yang and H. Miao, "Blind linguistic steganalysis against translation based steganography," in *Proc. IWDW*, Berlin, Germany, pp. 251–265, 2010.

[14] Z. Chen, L. Huang, H. Miao, W. Yang and P. Meng, "Steganalysis against substitution-based linguistic steganography based on context clusters," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 1071–1081, 2011.

[15] L. Xiang, X. Sun, G. Luo and B. Xia, "Linguistic steganalysis using the features derived from synonym frequency," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1893–1911, 2014.

[16] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv: 1408. 5882, 2014.

[17] Z. Yang, N. Wei, J. Sheng, Y. Huang and Y. J. Zhang, "TS-CNN: Text steganalysis from semantic space based on convolutional neural network," arXiv preprint arXiv: 1810. 08136, 2018.

[18] J. Wen, X. Zhou, P. Zhong and Y. Xue, "Convolutional neural network based text steganalysis," *IEEE Signal Processing Letters*, vol. 26, no. 3, pp. 460–464, 2019.

[19] Z. Yang, Y. Huang and Y. J. Zhang, "A fast and efficient text steganalysis method," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 627–631, 2019.

[20] Y. Niu, J. Wen, P. Zhong and Y. Xue, "A Hybrid R-BILSTM-C Neural Network Based Text Steganalysis," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1907–1911, 2019.

[21] L. Xiang, G. Guo, J. Yu, V. S. Sheng and P. Yang, "A convolutional neural network-based linguistic steganalysis for synonym substitution steganography," *Mathematical Biosciences and Engineering*, vol. 17, no. 2, pp. 1041–1058, 2020.

[22] Y. J. Bao, Y. Hao, Z. L. Yang and L. Sheng and Y. F. Huang, "Text Steganalysis with Attentional LSTM-CNN," in *Proc. ICCCS*, Shanghai, China, pp. 138–142, 2020.

[23] H. Yang, Y. J. Bao, Z. Yang, Y. F. Huang and S. M. Jiao, "Linguistic steganalysis via densely connected LSTM with feature pyramid," in *Proc. MM&Sec*, New York, NY, USA, pp. 5–10, 2020.

[24] Y. W. Zhang, W. M. Zhang, K. J. Chen, J. Y. Liu, Y. J. Liu *et al.*, "Adversarial examples against deep neural network based steganalysis," in *Proc. MM&Sec,* New York, NY, USA, pp. 67–72, 2018.

[25] W. Tang, B. Li, S. Tan, M. Barni and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, 2019.

[26] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv: 1609. 02907, 2016.

[28] Z. H. Huang, W. Xu, K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.

[29] C. Szegedy, W. Zaremba, I. Sutskever, I. Sutskever, J. Bruna *et al.*, "Intriguing properties of neural networks," arXiv preprint arXiv: 1312. 6199, 2013.

[30] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv: 1412. 6572, 2014.

[31] A. Nguyen, J. Yosinski and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. CVPR*, Boston, MA, USA, pp. 427–436, 2015.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv: 1706. 06083, 2017.

[33] W. Hamilton, Z. Ying and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. NIPS*, Long Beach, CA, USA, pp. 1024–1034, 2017.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio *et al.,* “Graph attention networks,” arXiv preprint arXiv: 1710. 10903, 2017.

[35] Z. Wu, S. Pan, F. Chen, G. Long, S. Y. Philip *et al.,* “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[36] W. Zaremba, I. Sutskever, O. Vinyals, “Recurrent neural network regularization,” arXiv preprint arXiv:1409.2329, 2014.

[37] S. Lahiri, “Complexity of word collocation networks: A preliminary structural analysis,” arXiv preprint arXiv: 1310. 5111, 2013.