ConTextING: Granting Document-Wise Contextual Embeddings to Graph Neural Networks for Inductive Text Classification

Yen-Hao Huang[†] and **Yi-Hsin Chen**[‡] and **Yi-Shin Chen**[*]
Institute of Information Systems and Applications[†*]
Department of Computer Science[*]
National Tsing Hua University, Hsinchu, Taiwan
Dcard, Taipei, Taiwan[‡]
{yenhao0218[†], eunicebes[‡], yishin[*]}@gmail.com

Abstract

Graph neural networks (GNNs) have been recently applied in natural language processing. Various GNN research studies are proposed to learn node interactions within the local graph of each document that contains words, sentences, or topics for inductive text classification. However, most inductive GNNs that are built on a word graph generally take global word embeddings as node features, without referring to document-wise contextual information. Consequently, we find that BERT models can perform better than inductive GNNs. An intuitive follow-up approach is to enrich GNNs with contextual embeddings from BERT, yet there is a lack of related research. In this work, we propose a simple yet effective unified model, coined *ConTextING*, with a joint training mechanism to learn from both document embeddings and contextual word interactions simultaneously. Our experiments show that ConTextING outperforms pure inductive GNNs and BERT-style models. The analyses also highlight the benefits of the sub-word graph and joint training with separated classifiers.

1 Introduction

Recently, the methods of non-sequential text modeling have gained attention, particularly for graph neural networks (GNNs) that learn document representation from graph structures. Most GNNs (Yao et al., 2019; Liu et al., 2020; Lin et al., 2021) are transductive since they are designed and built on a single heterogeneous graph, which connects all of the documents and words, including the training and testing data. Since testing documents must be used in training, transductive GNNs cannot be applied to new unseen documents. Thus, inductive learning GNNs (Huang et al., 2019; Nikolentzos et al., 2020; Zhang et al., 2020) have been proposed by representing each document in its own graph structure of local word interactions, with pretrained word embedding initialized on each word node. However, the global word embeddings are irrelevant to target documents, and graph structures might not capture the context well since text is usually produced in sequential order.

Modern transformer-based (Vaswani et al., 2017) pretrained models, such as BERT (Devlin et al., 2019), have shown their effectiveness in capturing context by sequentially modeling documents. In this study, we have also found that BERT-style models alone can outperform existing inductive GNNs. An intuitive method for enhancing inductive GNNs incorporates BERT contextual embeddings with GNNs for final text classification. For the most relevant research studies, Lu et al. (2020) adopted a GNN on a global vocabulary graph to enrich the token embeddings in BERT, and He et al. (2020) focused on sentences comparison tasks by feeding BERT contextual embeddings into a dependency graph. However, for text classification with inductive GNNs, there is a lack of reports on BERT and BERT-based GNNs. Only Lin et al. (2021) have adopted BERT's document embeddings for transductive GNNs.

Motivated by the recent success of inductive GNNs and the strengths of pretrained BERT models, in this work, we further consider the fact that these two types of models have different objectives. The former focuses on learning local syntactic word interactions, and the latter captures the context-aware semantics of words. To collaboratively join GNN and BERT models, we propose a unified model for learning *contextual inductive document representation via graph neural networks*, coined *ConTextING*, where each model has its own classifier for its own objective. A sub-word graph is adopted in ConTextING to focus more on fine-grained syntactic word usages, such as pre-/post-fix characters, which avoid over-focusing on content-specific word usages but maintain the flexibility in

[*] The corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1163–1168
October 12–17, 2022.

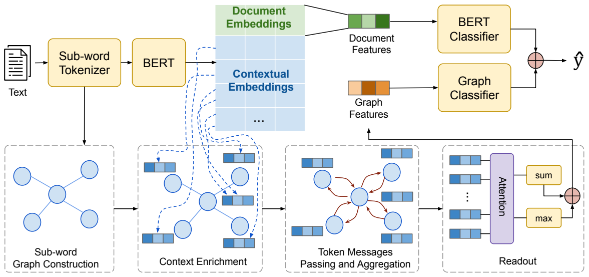


Figure 1: Overall framework of ConTextING.

accommodating to new words.

With this study, we make the following contributions: (1) We are the first to highlight the fact that pure GNNs are not superior to BERT and to provide detailed comparisons of state-of-the-arts (SOTAs), as well as the model variants. (2) We present ConTextING, a simple yet effective unification of BERT and GNN that yields results superior to pure inductive GNNs and BERT on a wide range of datasets. (3) We provide few-shot settings to illustrate the model's robustness to unseen words because of the sub-word adaptation.

2 ConTextING

This work proposed an unified model that consists of a BERT and a GNN modules as shown in Figure 1. ConTextING seamlessly enriches document-wise contextual information from a BERT-style model to the inductive GNN and makes final predictions based on the decisions of the two modules.

2.1 BERT-style Document Encoder

Given a text document, it is first tokenized into a sequence of sub-word tokens $\mathcal{T} = \{t_i\}$, and fed into the BERT-style model to obtain its document embeddings, $\mathbf{X} \in \mathbb{R}^{\delta}$ (from `[CLS]` token), and contextual embeddings, $\check{\mathbf{X}} \in \mathbb{R}^{|\mathcal{T}| \times \delta}$ for its tokens $\mathcal{T}$, where i dentoes the i-th sub-word in the document, and δ represents the hidden dimension of the BERT-style model.

Compared with the conventional GNNs, which utilize pretrained word embeddings (e.g. GloVe), the adaptation of contextual embeddings can capture local meanings within each document.

2.2 Sub-word Graph Construction

In contrast to previous GNNs, ConTextING constructs graphs from smaller word units—that is, the sub-word tokens—to capture more fine-grained text clues, such as the pre-/post-fix details of word usages. Such design can reduce the influence of topic-sensitive words and achieve robustness to rare words (Sennrich et al., 2016). For the sub-word graph, the sub-words are tokenized, based on byte-pair encoding (Sennrich et al., 2016) or WordPiece (Schuster and Nakajima, 2012) algorithms, according to the document encoder. The sub-word graph is formally defined in the following manner:

Definition 1. (Sub-word Graph) A sub-word graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V} \in \mathcal{T}$ represent unique sub-words and edges $\mathcal{E}$ are co-occurrences between sub-words.

The co-occurrences describe the preferences for word usages within the given document, which are obtained by a fixed-length sliding window on the sequence of sub-word tokens $\mathcal{T}$. The connectivity of the sub-word graph is calculated, following the work of Yao et al. (2019) as in Definition 2.

Definition 2. (Sub-word Connectivity) Let (v_i, v_j), $\mathbf{A}$ denote two linked sub-word nodes and the adjacency matrix of graph $\mathcal{G}$, respectively. The weight of this linked edge $\mathbf{A}_{i,j}$ is given by

$$\mathbf{A}_{i,j} = \begin{cases} \mathrm{PMI}(i,j), & v_i \neq v_j, \mathrm{PMI}(i,j) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathrm{PMI}(i,j) = \log \frac{p(i,j)}{p(i)p(j)}$ denotes the *point-wise mutual information*, $p(i), p(i)$ signify the probabilities of the sub-words' occurrence in all sliding windows, and $p(i,j)$ represents the probability of two sub-words' co-occurrence.

2.3 Context Enrichment

Given the fact that $|\mathcal{V}| \leq |\mathcal{T}|$ for a document, in order to jointly learn word interactions with contextual information in a graph view, it is necessary to define a mapping matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{T}|}$ for converting the features of sub-word t_j to node v_i by

$$\mathbf{M}_{i,j} = \begin{cases} 1/\mathrm{freq}(v_i), & v_i = v_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathrm{freq}(v_i)$ denotes the occurrences of each node v_i in $\mathcal{T}$. The contextual node representation $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times \delta}$ is then retrieved by $\mathbf{H} = \mathbf{M}\check{\mathbf{X}}$.

2.4 Token Messages Passing and Aggregation

To learn the word interactions, a token message passing and aggregation (TMPA) step is adopted. With the success of the gated structure and attention mechanism in natural language processing, in this work, we simply adopted gated graph recurrent units (Li et al., 2015) as TextING (Zhang et al., 2020) and the graph attention network (GAT) (Veličković et al., 2018) on sub-word graph $\mathcal{G}$. The adoption of other graph convolution methods (Kipf and Welling, 2017; Hamilton et al., 2017) are left for future works.

Formally, the value of a node v can be updated by aggregating the information $\mathbf{H}_{\mathcal{N}}^{(v)} \in \mathbb{R}^{|\mathcal{N}|\times\delta}$ from its 1-hop neighbors and its current value $\mathbf{H}^{(v)} \in \mathbb{R}^{\delta}$, where $|\mathcal{N}|$ denotes the number of neighbors for node v. One updating process refers to one TMPA operation for sub-word interactions, based on the study of Zhang et al. (2020). By stacking the TMPA for τ times, each node can obtain features from neighbors within the τ-hop distance. The node representation after τ TMPA is denoted by $\mathbf{H}^{(v,\tau)}$.

2.5 Graph Readout and Jointly Learning

A graph readout step is applied to aggregate the final node embeddings in order to obtain a graph-level document representation $\mathbf{H}^{(G)}$ as follows:

$$\widehat{\mathbf{H}}^{(v)} = \sigma(f_1(\mathbf{H}^{(v,\tau)})) \odot \tanh(f_2(\mathbf{H}^{(v,\tau)})) \quad (3)$$

$$\mathbf{H}^{(G)} = \frac{1}{|\mathbb{V}|}\sum_{v\in\mathbb{V}} \widehat{\mathbf{H}}^{(v)} + \text{Maxpooling}(\widehat{\mathbf{H}}) \quad (4)$$

where f_1 and f_2 represent two dense layers for the weighted embeddings of each sub-word node through a soft attention mechanism by a sigmoid function σ. The average summation and feature-wise max-pooling functions are subsequently applied to obtain graph representation $\mathbf{H}^{(G)}$.

To this end, two features are extracted: the sequential document features $\mathbf{X}$ from the BERT module and the non-sequential features $\mathbf{H}^{(G)}$ from the GNN module. It is worth noting that in this work, we consider that the aims of BERT and the GNN are essentially different (sequential and non-sequential modeling, respectively). A directly concatenating, adding, or averaging operation $\mathbf{X}$ and $\mathbf{H}^{(G)}$ may blur the distinction between the objectives of these two models. We thus adopt two classifiers separately for each model with a linear interpolation (Lin et al., 2021) to regulate the objectives of BERT and GNN for final prediction.

$$\hat{y}^{(\text{BERT})} = \text{softmax}(\mathbf{W}_x\mathbf{X} + \mathbf{b}_x) \quad (5)$$

$$\hat{y}^{(\text{Graph})} = \text{softmax}(\mathbf{W}_g\mathbf{H}^{(G)} + \mathbf{b}_g) \quad (6)$$

$$\hat{y} = (1-\eta)\hat{y}^{(\text{BERT})} + \eta\hat{y}^{(\text{Graph})} \quad (7)$$

where $\eta \in [0,1]$ denotes a hyper-parameter to decide the main objective between BERT and GNN. A higher η value indicates the more focuses on non-sequential word interactions. The use of the automatic mechanism to determine η is left for our future works.

3 Experiments

The proposed model is evaluated by addressing these concerns: (1) Is ConTextING better than pure BERT-style models and inductive GNNs for text classification? (2) Can the ConTextING achieve satisfactory results with limited training data?

Datasets. Five common benchmark datasets for evaluating GNNs are adopted and pre-processed, following the works of Yao et al. (2019); Zhang et al. (2020); Lin et al. (2021), which are medical abstracts with 23 diseases classes (Ohsumed); movie reviews (MR) with sentiment polarities; Reuters newswire items with 8 (R8) and 52 (R52) categories; and 20NewsGroups (20NG) with 20 categories, respectively.

Baselines. The compared baselines include (1) *traditional deep learning models* with GloVe embeddings (Pennington et al., 2014): textCNN (Kim, 2014), LSTM (Liu et al., 2016) and bi-directional LSTM (Bi-LSTM); (2) *SOTA language models*: BERT (BT) (Devlin et al., 2019) and RoBERTa (RBT) (Liu et al., 2019); (3) *SOTA inductive GNNs*: TextGCN-*ind* (an inductive version by Yao et al. (2019)), text-level GNN (Huang et al., 2019), TextING (Zhang et al., 2020), and HyperGAT-*ind* (Ding et al., 2020) (topics are learned without testing data.); and (4) GNN-enriched BERT classifier: VGCN-BERT (Lu et al., 2020).

Since the codes released by HyperGAT's authors include testing data when producing its topic features, HyperGAT-*ind* is then reproduced by excluding the testing data when generating the topics. VGCN-BERT is also reproduced as its authors only reported the F1 score on MR. All of the reproduced results are based on the original authors' codes[1] and the parameters described in the original papers.

The results of some baselines are obtained from Zhang et al. (2020); Ding et al. (2020) for a fair comparison. Note that they both followed the same setting and their baseline results are obtained from the work by Yao et al. (2019).

Experimental Settings. ConTextING consists of a base version of BT/RBT and a two-layer gated

[1] https://github.com/kaize0409/HyperGAT; https://github.com/Louis-udm/VGCN-BERT; https://github.com/CRIPAC-DIG/TextING

Method	**Ohsumed**	**MR**	**R8**	**R52**	**20NG**
CNN+GloVe	58.44	77.75	95.71	87.59	82.15
LSTM+GloVe	51.10	77.33	96.09	90.48	75.43
Bi-LSTM+GloVe	49.27	77.68	96.31	90.54	73.18
BERT (BT)	68.74	85.88	97.26	96.26	84.54
RoBERTa (RBT)	69.86	87.08	97.35	95.48	84.07
TextGCN-*ind*	57.70	74.80	95.78	88.20	83.31
Text-level GNN	69.40	75.47	97.89	94.60	84.16
TextING	70.42	79.82	98.04	95.48	82.48
HyperGAT-ind	67.33	77.08	97.03	94.55	84.63
VGCN-BERT	70.19	85.93	97.89	95.87	55.76
ConTextING-BT	71.28	86.01	97.91	**96.52**	**86.19**
w. GAT-BT	71.51	86.16	97.96	96.28	**86.25**
ConTextING-RBT	**72.53**	**89.43**	**98.13**	**96.40**	85.00
w. GAT-RBT	**72.06**	**89.24**	**98.09**	96.15	84.97

Table 1: Test accuracy comparison of inductive methods. The results of the methods highlighted in *italics* are produced by this work.

Method	**MR**	**Ohsumed**
TextGCN	53.15 (-23)	47.24 (-21)
TextING	64.43 (-15)	*51.40 (-19)*
RBT	69.16 (-18)	50.51 (-19)
ConTextING-RBT	**73.14** (-16)	**53.67** (-19)
# Samples/words in Training	40/465	*448/7,009
# New Words in Test	18,299	7,148

Table 2: Test accuracy under a few-shot setting. Values in parentheses are performance reductions from Table 1.

graph recurrent unit with 768 hidden size (800 for the GAT variant). The window size, η, dropout, learning rate, loss function and train epochs are set as 3 (widely used in GNNs), $0.9/0.3$ (MR/others), 0.5, $1e^{-5}$, negative log likelihood, and 80, respectively. All results are averaged over 10 runs.

3.1 Benchmark Text Classification

Test Performance. Table 1 summarizes the test accuracy of each model. Overall, it can be observed that ConTextING (four different variants in total) generally beats all of the baselines, including the SOTA models, on every dataset. This indicates the benefits of integrating text modeling in both sequential and non-sequential manners. By concatenating word embeddings from a global vocabulary graph (VGCN) to BERT, VGCN-BERT also performs well on these benchmark datasets, except for the 20NG[2]. The low accuracy of 20NG might be caused by the unbalanced distribution of token embeddings between BERT and the graph embeddings produced by the VGCN component due to a highly sparse vocabulary graph from a large amount of vocabulary ($> 25k$).

Performance Boost Over Pure BERT and GNNs. Compared with pure BT, RBT, and GNNs, the results reveal that ConTextING consistently obtains $1 - 2.7$ points of gains on accuracy from pure BT and RBT for all datasets. Regarding GNNs, it is discovered that pure GNNs are not superior to BERT-style models. Similar boosts are also observed, particularly for the MR dataset, which improves accuracy by approximately 9 points from SOTA GNN (TextING). Such significant gains are mostly contributed by the BERT module, as BT and RBT themselves can obtain high accuracy with the merits of large-scale pretraining. In contrast, on Ohsumed, although BT/RBT are beaten by inductive GNNs, ConTextING-RBT can still obtain a high score of 72.53 on accuracy.

3.2 Few-shot Inductive Capability

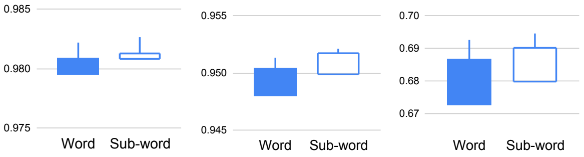
To examine ConTextING's inductive capability, we conduct few-shot learning experiments on benchmark datasets according to the setting by Zhang et al. (2020). The number of training samples is limited to a maximum of 20 labeled documents per class. Consequently, most words in the test set are unseen in these settings. The results are compared with those of TextGCN and TextING reported by Zhang et al. (2020)[3]. The results presented in Table 2 show that our model is the most robust one among a few training samples. The observations are basically aligned with the results in Table 1. For MR, the RBT alone could perform better than baselines. Although RBT has a worse result than the one by TextING on Ohsumed, ConTextING could further boost the performance of RBT on both datasets with the aid of the GNN module. By taking sub-words, which are naturally robust to the new words, as input for BERT and GNN modules, ConTextING is thus more stable.

3.3 Model Analysis

Word and Sub-word Graph Comparison. To evaluate the effectiveness of the sub-word graph, we adapt TextING but modify its input graph into a word graph and a sub-word graph with the fixed fine-tuned RBT's embeddings with window size 3 (best for a word graph). Figure 2 shows that TextING with a sub-word graph is able to perform better consistently on different benchmark datasets.

[2] The reproduced *VGCN-BERT* results are consistent with the reported values in the original paper. By initializing *VGCN-BERT* with a pretrained BT, it can obtain 60 on the accuracy of 20NG. However, its accuracy decreases after the first epoch.

[3] Note that there are only nine samples in the training set for the "C22" class on Ohsumed (which differs from those in the original report); thus, TextING is reproduced by using the original authors' codes.



(a) R8 (b) R52 (c) Ohsumed

Figure 2: Accuracy of using word and sub-word graph.

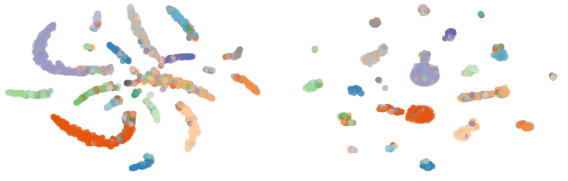
Model Variants (Acc. ± Std.)	**R52**	**Ohsumed**	**20NG**
ConTextING-RBT	96.4±.2	72.5±.3	85.0±.4
(i) w/o. joint train (concat. Embd.)	96.2±.2	72.4±.7	84.4±.5
(ii) w/o. RBT classifier	95.8±.5	72.2±.6	84.5±.3
(iii) w. fix RBT Word Embd.	95.2±.1	66.7±.6	81.2±.4
(iv) w. fix tuned RBT Word Embd.	94.9±.2	68.0±.8	83.9±.4

Table 3: Ablation studies on ConTextING variants.

Effects of Joint Training. To examine the effectiveness of joint training, common methods for aggregating the BERT-style model (RBT) and GNN are implemented, with careful inspections of different hyper-parameters. Table 3 shows the superior performance of ConTextING with joint training. With RBT updated during training, (i) and (ii) can obtain high accuracy under a low learning rate; however, it is still slightly worse than adopting joint training. As for MR, comparable results could be achieved without the joint training.

For (iii) and (iv), the effects of RBT's contextualized embeddings are examined, where the embeddings are fed into the pure GNNs module of ConTextING without training the RBT. In other words, ConTextING is simplified as TextING architecture, with its node features initialized as contextualized embeddings. The results show that the contextualized embedding by the fine-tuned RBT improves by 1.3 and 2.7 points on Ohsumed and 20NG, respectively, over the RBT embedding without finetuning. It is also observed that (iii) and (iv) perform worse than TextING with its original GloVe embeddings, which shows that GNNs alone may be unable to process the high-dimension embeddings well (i.e. 300 v.s. 768). Similarly, the unification in (iii) is also found easily-fail-to-converge (60%) on MR.

BERT and GNN Embedding Comparison. To indicate the difference in what BERT and GNN modules have learned, t-SNE (Van der Maaten and Hinton, 2008) is applied to visualize the corresponding document features in ConTextING-RBT. Figure 3 reveals that the GNN module produces a representation different from RBT's one. Specifically, the RBT module tends to mess up several documents (center of Figure 3a), while the GNN module can distinguish them more correctly. By jointly training and predicting on *two different classifiers*, ConTextING could achieve superior performance than each of them individually.



(a) RoBERTa (b) GNN

Figure 3: Visualizations of BERT and GNN modules in ConTextING on Ohsumed's test documents. The color of a node corresponds to the node's class.

4 Conclusion

In this paper, we have proposed ConTextING, which successfully learns document embeddings sequentially and contextual word interactions non-sequentially at the same time. Various context encoders or GNNs are also allowed to build ConTextING on top of this framework. In the future, we aim to involve GNNs in a large-scale pretraining process in combination with BERT.

Acknowledgements

This research is supported by the National Science and Technology Council in Taiwan (NTSC 111-2221-E-007-110-MY3; NTSC 110-2221-E-007-085-MY3; NTSC 108-2221-E-007-064-MY3).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4927–4936.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.

Qi He, Han Wang, and Yue Zhang. 2020. Enhancing generalization in natural language inference by syntax. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4973–4978, Online. Association for Computational Linguistics.

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3444–3450.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *International Conference on Learning Representations*.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining gnn and bert. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *IJCAI*.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8409–8416.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. Vgcn-bert: Augmenting bert with graph embedding for text classification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 369–382. Springer.

Giannis Nikolentzos, Antoine Tixier, and Michalis Vazirgiannis. 2020. Message passing attention networks for document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8544–8551.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. ACL.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 334–339.