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‡
{yenha0218†, 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 pre-trained 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...
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 \( T = \{t_i\} \), and fed into the BERT-style model to obtain its document embeddings, \( \delta \in \mathbb{R}^{|T|} \) (from [CLS] token), and contextual embeddings, \( \tilde{X} \in \mathbb{R}^{T \times |T|} \) for its tokens \( T \), where \( i \) denotes the \( i \)-th sub-word in the document, and \( \delta \) 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 \( \mathcal{G} = (V, E) \) where vertices \( V \subseteq T \) represent unique sub-words and edges \( E \) 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 \( T \). 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 \((v_i, v_j)\), \( A \) denote two linked sub-word nodes and the adjacency matrix of graph \( \mathcal{G} \), respectively. The weight of this linked edge \( A_{i,j} \) is given by

\[
A_{i,j} = \begin{cases} 
\text{PMI}(i, j), & v_i \neq v_j, \text{PMI}(i, j) > 0 \\
0, & \text{otherwise} 
\end{cases}
\]  

(1)

where \( \text{PMI}(i, j) = \log \frac{p(i,j)}{p(i)p(j)} \) denotes the pointwise mutual information, \( p(i), p(j) \) 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 \( |V| \leq |T| \) 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 \( M \in \mathbb{R}^{|V| \times |T|} \) for converting the features of sub-word \( t_j \) to node \( v_i \) by

\[
M_{i,j} = \begin{cases} 
1/\text{freq}(v_i), & v_i = v_j \\
0, & \text{otherwise} 
\end{cases}
\]  

(2)

where \( \text{freq}(v_i) \) denotes the occurrences of each node \( v_i \) in \( T \). The contextual node representation \( H \in \mathbb{R}^{|V| \times \delta} \) is then retrieved by \( H = MX \).

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 Text-
ing (Zhang et al., 2020) and the graph attention
network (GAT) (Veličković et al., 2018) on sub-
word graph $G$. The adoption of other graph con-
volution 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 $H^{(v)} \in \mathbb{R}^{|N| \times d}$ from
its 1-hop neighbors and its current value $H^{(v)} \in \mathbb{R}^d$, where $|N|$ 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 $\tau$ times, each node can obtain features from
neighbors within the $\tau$-hop distance. The node
representation after $\tau$ TMPA is denoted by $H^{(v, \tau)}$.

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 $H^{(G)}$ as follows:

$$
\hat{H}^{(v)} = \sigma(f_1(H^{(v, \tau)})) \odot \tanh(f_2(H^{(v, \tau)})) \\
H^{(G)} = \frac{1}{|V|} \sum_{v \in V} \hat{H}^{(v)} + \text{Maxpooling}(\hat{H})
$$

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 $\sigma$. The average summation and feature-
wise max-pooling functions are subsequently ap-
plied to obtain graph representation $H^{(G)}$.

To this end, two features are extracted: the se-
quential document features $X$ from the BERT mod-
ule and the non-sequential features $H^{(G)}$ 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 di-
rectly concatenating, adding, or averaging opera-
tions $X$ and $H^{(G)}$ 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.

$$
\hat{y}^{(\text{BERT})} = \text{softmax}(W_{x}X + b_{x}) \\
\hat{y}^{(\text{Graph})} = \text{softmax}(W_{g}H^{(G)} + b_{g}) \\
\hat{y} = (1 - \eta)\hat{y}^{(\text{BERT})} + \eta\hat{y}^{(\text{Graph})}
$$

where $\eta \in [0, 1]$ denotes a hyper-parameter to de-
cide the main objective between BERT and GNN.

A higher $\eta$ value indicates the more focuses on
non-sequential word interactions. The use of the
automatic mechanism to determine $\eta$ 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 medi-
cal 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 em-
bbeddings (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 with-
out 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 fea-
tures, HyperGAT-ind is then reproduced by exclud-
ing 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’ codes1
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

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

\[\text{VGCN-BERT}.\]

**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.

\[\text{Note that there are only nine samples in the training set for the “C22” class on Ohsumed (which differs from those in the original report); thus, TextING is reproduced by using the original authors’ codes.}\]
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.

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


