EasyChair Preprint

№ 9135

Comparative Study of Inductive Graph Neural Network Models for Text Classification

Saran Pandian, Uttkarsh Chaurasia, Shudhanshu Ranjan and Shefali Saxena

October 26, 2022

COMPARATIVE STUDY OF INDUCTIVE GRAPH NEURAL NETWORK MODELS FOR TEXT CLASSIFICATION

Saran Pandian[1], Uttkarsh Chaurasia[2], Shudhanshu Ranjan[3], and Shefali Saxena[4]

[1]DAIICT,Gandhinagar
[2]Department of Computer Science and Engineering, NIT Hamirpur
[3]Presidency University, Yelahanka, Banglore
[4]Department of Electronics and Communication, NIT Hamirpur

***Abstract*—Among several proposed methods for text classification, transformers and GNN have gained popularity recently. Models which use GNN are both transductive and inductive. Transductive models such as TextGCN fail to deal with scalability issues with larger datasets because of converting the whole corpus into a graph. Induction models were introduced, which convert individual documents to graphs fed to the model for classification. In this paper, a comparative study of the three Inductive Graph Neural Network(GNN), namely TexTING, In-GCN, In-GAT models, is analyzed. The study shows that In-GAT gave better result comparecd other two models. Also, It is proved that message passing mechanism does not have effect on performance of model and Entropy loss value depends on size of Dataset and Model used.**

***Index Terms*—Text Classification, Graph Attention Network, Gated Graph Recurrent Unit, Entropy, Inductive Model, Graph Convolutional Network.**

I. Introduction

Text classification, one of the fundamental tasks of NLP, has a wide variety of applications in the real world. Spam detection, document classification, sentiment analysis are some of them. Recently, text classification techniques based on Graph neural networks and Transformers[1] have gained popularity due to their robustness in handling complex data and ability to handle contextual representation of words. Transformers with Self-attention mechanism[1] can provide very high accuracy at the cost of higher computational costs. On the other hand, GNNs can give better results with less computational costs.

To deal with data generated from non-Euclidean domains, GNNs were introduced, represented as graphs with complex relationships between nodes as given in Fig 1(a) where each node is connected different layers of nodes. It has wide applications in Node Classification[2], Link prediction[3], Graph classification[4], and many more. Data have complex structures. In chemistry, GNNs are used to research the graph structure of molecules and compounds[5]. Most widely used GNN models are Gated Graph Recurrent Unit(GGRU) [6], Graph Attention Network(GAT) [7] and Graph Convolutional Networks(GCN) [2] each having its own advantages.

GNNs have application in text classification problems in NLP. There are two types of algorithms, namely transductive and inductive algorithm [8]. Transductive algorithms such as TextGCN use a single graph with global relations between documents and words, whereas inductive algorithms use word-to-word relations and interactions. Transductive algorithms pose scalability issues, so they cannot be used for larger datasets. On the other hand, inductive algorithms such as TexTING [8] can be used for large datasets as it converts every document to graphs using a word co-occurrence matrix.

The follwing are the contributions of this paper:

- Comparison of peformance of three models of GNN for inductive text classification over three datasets, namely IMDB, AG news, DBpedia
- Study of the effect of the message passing mechanism.
- Study of the effect of Entropy loss Values with respect to model and size of dataset.

II. Previous works

Text-GCN [9] is a transductive model where a single text graph for the whole corpus is built based on word-word and document-word relations. TextGCN learns embeddings by applying GCN, which gives output embeddings for documents and words. This model uses document embeddings to predict the class of the document. However, this method is not feasible for larger datasets as it has scalability issues.

Text Level Graph Neural Network for Text Classification [10] proposes a new GNN based model that builds graphs for each document instead of the whole corpus. The study uses smaller windows over the text to build Graphs, which extract local information of the text. Thus, it removes the scalability issues of TextGCN. However, this method uses a global structure for word embeddings which inherently becomes a transductive model.

Text classification method for Inductive word representations via Graph neural networks [8] resolves this issue by creating a word co-occurrence matrix for each document, thus removing the necessity for a global structure. This method uses GGRU [6] to represent words in each document.

III. Methodology

A. Dataset Description

The dataset used in this study is the AG news dataset[1], the IMDB[2] movie reviews dataset and the DBpedia ontology classification dataset[3].

AGnews dataset is derived from AG's corpus of over 1 million news articles. The dataset is a multiclass classification dataset with four classes: World, Sports, Business, and Sci/Tech classes. It consists of 120,000 training data and 7,600 test data. Each category contains 30,000 training samples and 1,900 testing samples.

IMDB movie reviews dataset is a binary classification dataset that categorises movie reviews as either positive or negative. This dataset includes highly polarised reviews. Movie reviews with a rating of less than equal to 4 out of 10 are labelled as negative, while those with ratings of more than or equal to 7 are labelled as positive. It contains 25,000 data points for training and 25,000 data points for testing.

The DBpedia ontology classification dataset is a text dataset with 14 classes from DBpedia 2014. Each class represent individual news text strings. The total size of the training dataset is 560,000, and the testing dataset is 70,000.

The paper follows a similar methodology used in [8]. The steps are threefold: construction of graph, learning the graph embeddings and ReadOut Layer. The pipeline for the Inductive Text Classification techniques is given in Fig 1(b).

B. Construction of Graph

Each document is represented in Graph $G = (V, E)$, where V denotes the document's representation and E represents the co-occurrence of words in the corpus. The adjacency matrix and feature matrix are two matrices as input to the GNN models.

1) adjacency matrix: Since the inductive method is used for text classification, every document is converted to graphs rather than whole corpus. The co-occurrence matrix represents text as a set of co-occurrent words [11]. The co-occurrence matrix represents the relationship between words adjacent to each other within a fixed window size. The adjacency matrix is normalised as follows:

$$A_{normalised} = \tilde{D}_{ii}^{-\frac{1}{2}} A \tilde{D}_{ii}^{-\frac{1}{2}} \tag{1}$$

where

$$\tilde{D}_{ii} = \sum_{j} A_{ij} \tag{2}$$

For TexTING $A_{normalised}$ is used as an adjacency matrix. For In-GCN, since self connections are considered, the following adjacency matrix is used:

$$\tilde{A}_{normalised} = \tilde{D}_{ii}^{-\frac{1}{2}} \tilde{A} \tilde{D}_{ii}^{-\frac{1}{2}} \tag{3}$$

[1]http://groups.di.unipi.it/ gulli/AG_corpus_of_news_articles.html

[2]https://ai.stanford.edu/ amaas/data/sentiment/

[3]https://github.com/le-scientifique/torchDatasets/raw/master/dbpedia_csv.tar.gz

Where

$$\tilde{A} = A + I_N \tag{4}$$

Where $A \in R^{|V|\times|V|}$ is a binary matrix based on word co-occurrence and I_N denoting the identity matrix for self connections [2] where V is a vocabulary set of the sentence.

2) feature matrix: A feature matrix is created H$\in R^{|V|\times|d|}$ by using the glove embeddings of dimension d. This feature matrix gets enriched after passing it into the GNN model.

C. Learning the Graph Embeddings

Three GNN models were used to create graph embeddings, namely Gated Graph recurrent network(GGRU), Graph convolutional network(GCN), Graph attention network(GAT) for creating the word embeddings.

1) Gated Graph neural network: Gated Graph Neural Networks [6] is used to learn the embeddings of the word nodes. A node representation includes the neighbouring word representations and the current word representations thus representing a word using local context. The architecture for GGRU is inspired from the RNN variant Gated Recurrent Unit [12].

$$a^{t+1} = A_{normalised} H^t W_a \tag{5}$$

$$z^{t+1} = \sigma(W_z a^{t+1} + U_z H^t + b_z) \tag{6}$$

$$r^{t+1} = \sigma(W_r a^{t+1} + U_z H^t + b_z) \tag{7}$$

$$\tilde{H}^t = \tanh\left(W_h a^{t+1} + U_h(r^t \odot H^t) + b_z\right) \tag{8}$$

$$H^{t+1} = \tilde{H}^{t+1} \odot z^{t+1} + H^t \odot (1 - z^{t+1}) \tag{9}$$

Where W, U, b are learnable parameters. z and r represent the update gate and reset gate representation. a denotes representation of neighbour nodes. t represents the number of layers or hops the message is passed on in a graph.

2) Graph convolutional network: The information from t layer is transferred to $t+1$ layer using Graph convolution. The information about the neighbour nodes is transferred through the adjacency matrix $A_{normalised}$. H matrix denotes the feature matrix. W denotes the weight matrix, which is trainable.

$$H^{t+1} = \sigma(\tilde{A}_{normalised} H^{(t)} W^{(t)}) \tag{10}$$

3) Graph attention network: GAT has the same mechanism as GCN with attention mechanism added on it. Instead of using the normalised weights in $\tilde{A}_{normalised}$, attention mechanism α_{ij} is applied. Matrix $H^t = [\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_N]$ denotes the feature matrix where $\overrightarrow{h}_j$ is the node representation of $j \in \{1, ..., N\}$ th word of a document. The updated representation of node $\overrightarrow{h}_j$ for k-th head is given as

$$\overrightarrow{h}_j'^k = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} W^k \overrightarrow{h}_j\right) \tag{11}$$

To compute α_{ij} the attention setup inspired from Bahadanau multi-head attention [13] is used. e_{ij} that indicate the importance of node j's features to node i is computed.

$$e_{ij} = a(W\overrightarrow{h}_i \| W\overrightarrow{h}_j) \tag{12}$$

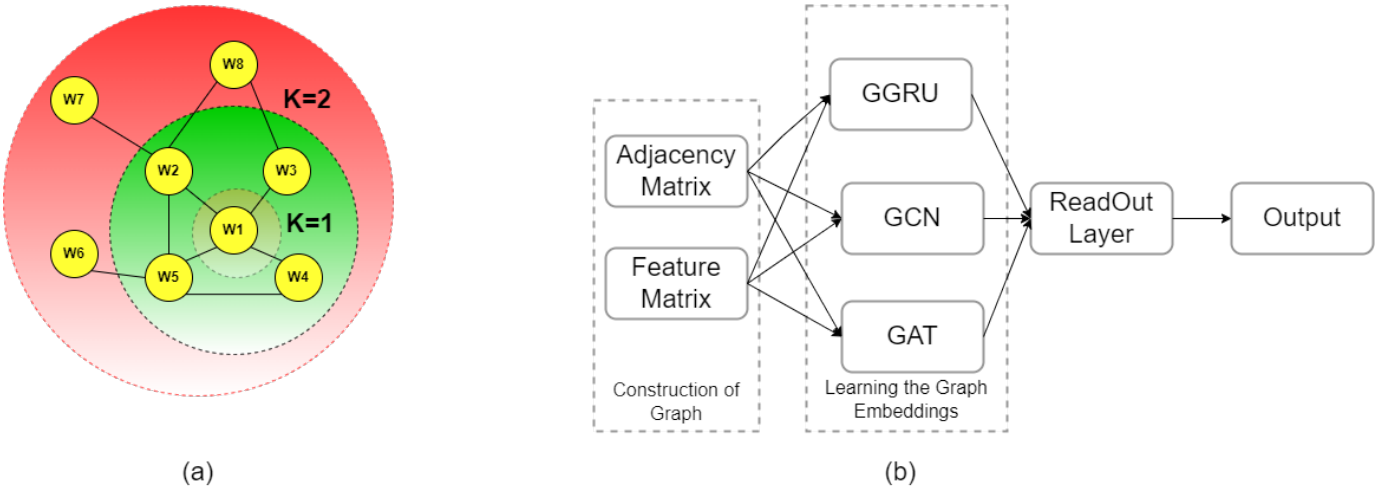

Fig. 1. (a) With respect to node w1, nodes w2, w3, w4 and w5 constitute layer 1 and w6, w7 and w8 constitute layer 2. (b) Pipeline for Inductive Text Classification Models.

Where $a \in \mathcal{R}^{2F'}$ contains weights for shared attention mechanism and $W \in \mathcal{R}^{F' \times F}$ matrix contains trainable parameters for computing attention weights.

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU(e_{ij})\right)}{\sum_{k \in \mathcal{N}_i} \exp(LeakyReLU(e_{ij}))} \tag{13}$$

$$\overrightarrow{h}'_j = \|_{k=1}^{K} \overrightarrow{h}'^{k}_j \tag{14}$$

$$\overrightarrow{h}'_j = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W^K \overrightarrow{h}_j \right) \tag{15}$$

For Multihead self attention with K heads, the output from different attention heads denoted by W^k are concatenated where k $\in 1...K$ and pass it to the next layer as h'.

$$H^{t+1} = [\overrightarrow{h}'_1, \overrightarrow{h}'_2, ..., \overrightarrow{h}'_N] \tag{16}$$

Here, H^{t+1}denotes the updated Feature Matrix where $\overrightarrow{h}_j$ represent updated node features.

D. ReadOut Layer

The same methodology used in TexTING [8] is followed for predicting the classes. The output of the GNNs (9,10,16) are passed to multilayer perceptrons (MLP) f_1 and f_2.

$$G_v = \sigma(f_1(H^{t+1})) \odot \tanh\left(f_2(H^{t+1})\right) \tag{17}$$

MLP function f_2 does non-linear transformation, whereas f_1 filters the necessary features for the prediction. In addition, Averaging and Maxpooling are done to include the contribution of all features and keyword features.

$$G_v = [g_1, g_2, ..., g_{\mathcal{V}}] \tag{18}$$

$$h_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} g_v + Maxpooling(g_1 ... g_{\mathcal{V}}) \tag{19}$$

since it is a classification problem softmax layer is used at the end and cross entropy as loss function. $\hat{\mathcal{Y}}_{\mathcal{G}}$ is the probability of the predicted class.

$$\hat{\mathcal{Y}}_{\mathcal{G}} = softmax(W h_{\mathcal{G}} + b) \tag{20}$$

$$\mathcal{L} = -\sum_{i} \mathcal{Y}_{\mathcal{G}_i} \log \hat{\mathcal{Y}}_{\mathcal{G}} \tag{21}$$

$\mathcal{L}$ denotes the loss value which is to be optimised.

IV. Experiments

A. Evaluation Metrics

Since the dataset is perfectly balanced, the study uses accuracy for evaluation. Train-test split was available in the dataset.

B. Parameter Settings

All the three models are implemented using PyTorch. The dimension of the Glove embedding is fixed to 300. The optimisation is done using Adam Optimizer with a batch size of 16. Glorot initialisation [14] is used for initialising model parameters. The dropout ratio is tuned to 0.5 and the learning rate to 0.0005. Window size is set to 3 for all three models. Finally, the number of heads parameter is tuned to 16 for In-GAT.

V. Results and Observations

A. Effect of Message Passing Mechanism

The message passing mechanism is used to enrich the embeddings of the nodes(words)by allowing message passing among nodes at k nodes away. Thus the final embeddings of the nodes contain the information about self-nodes and nodes within k steps. Fig 1(a) shows how layers are defined for a graph.

Fig 2 shows the training loss curve for each combination of model and dataset for different interaction step value k.

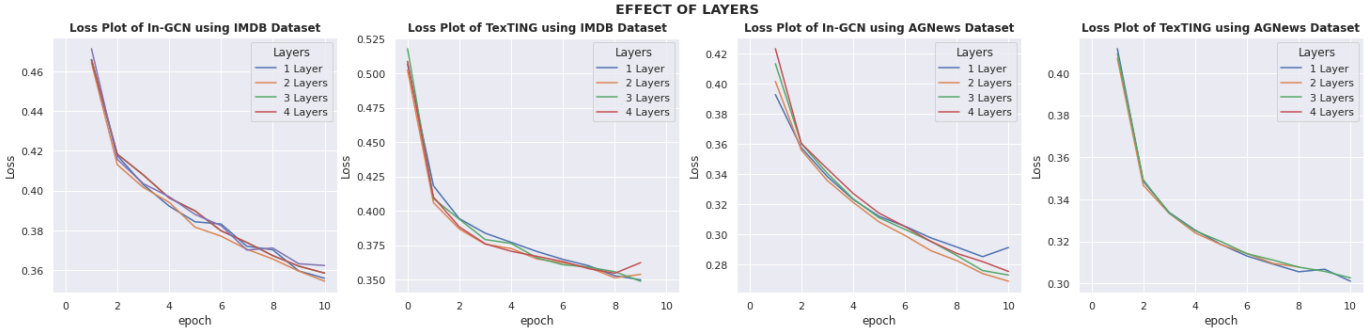

Fig. 2. The loss curves show that the convergence of loss function for each model is almost similar irrespective of the number of layers

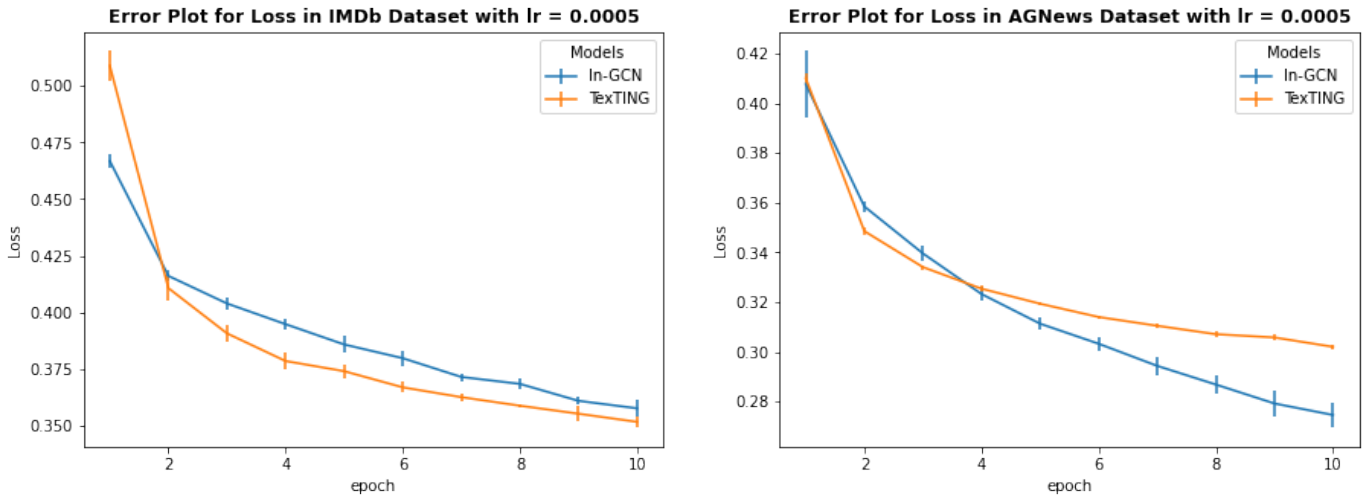

Fig. 3. The mean and standard deviation of loss values for different layers at each epoch is plotted. The loss values can be seen in Fig 2

As observed from Fig 3, which shows the mean and standard deviation of loss values for each epoch, the effect of the message passing mechanism is negligible in contrast to TextGCN [9] where the effect was observed for the first order and second order interactions. [8] showed that with increasing the interaction steps of message passing, accuracy could be increased up to interaction steps of 2 or 3. However, this effect is negligible, with an increase of 0.1% to 1%. A higher standard deviation value observed is 0.0133 for In-GCN on AGNews Dataset. The negligible standard deviation values show that the significance of the message passing mechanism is negligible.

B. Entropy Of The Output

Entropy is the measure of randomness in prediction by a Machine Learning model. The primary goal of the Classifier model is to reduce the entropy of a model to come up with maximum accuracy for output. Higher entropy means a higher loss value.

As observed from the Fig 3, for a fixed learning rate, the entropy at the output layer is higher for more complex models such as In-GAT when applied over smaller datasets such as IMDB at the initial stages of training. This is evident from the high training loss for the first epoch. For larger datasets such as DBPedia, the training loss at the beginning stage is less because the dataset is large enough to reduce the entropy for the first epoch. As a result, the In-GAT model's convergence is faster than simple models for all three datasets.

C. Performance of the Models

The study uses accuracy for measuring the performance of the models. Overall, In-GAT gave better results than the other two models, as shown in Table I. The reason is the use of attention mechanism in In-GAT, which gives more weightage for the most relevant parts of the input to make decisions.[7]. Also this attention mechanism helps in interpretability.

TABLE I
ACCURACY VALUES OF GNN MODELS OVER DIFFERENT DATASETS

	TexTING	In-GCN	In-GAT
IMDB	0.86666	0.86616	**0.87735**
AGnews	0.91118	0.91276	**0.91486**
DBpedia	0.96667	0.97971	**0.981071**

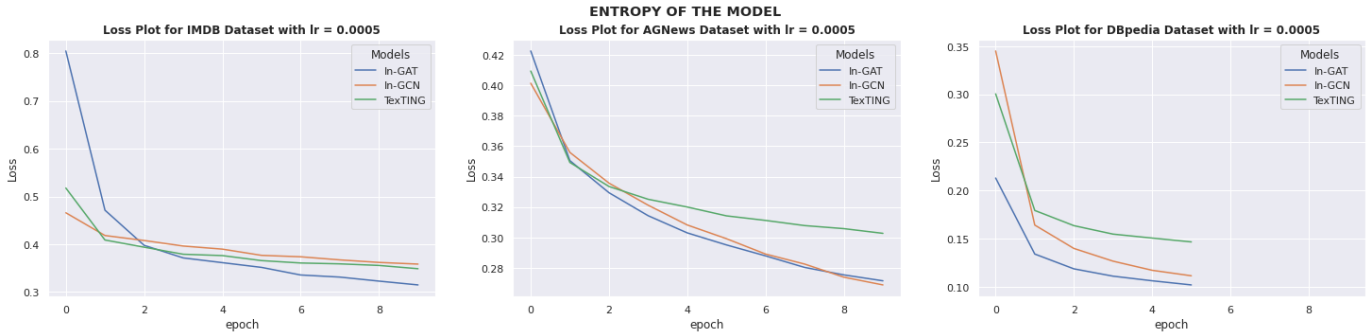

Fig. 4. The loss curves show that the training loss for In-GAT is higher for the first epoch over smaller dataset such as IMDB and lesser for larger dataset such as DBPedia.

VI. Conclusion and Future Work

After working on combinations of different datasets and models we proved that In-GAT to be the best model, especially with a larger dataset.

This paper is a comparative analysis of different models. Future work can focus on pushing the current results to the state of the art results. This model can also include a global attention mechanism with a different adjacency matrix instead of the co-occurrence matrix used here.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[3] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[4] Y. Wang, W. Wang, Y. Liang, Y. Cai, and B. Hooi, "Graphcrop: Subgraph cropping for graph classification," *arXiv preprint arXiv:2009.10564*, 2020.

[5] M. Tsubaki and T. Mizoguchi, "On the equivalence of molecular graph convolution and molecular wave function with poor basis set," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1982–1993, 2020.

[6] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[8] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," *arXiv preprint arXiv:2004.13826*, 2020.

[9] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 7370–7377.

[10] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3444–3450. [Online]. Available: https://aclanthology.org/D19-1345

[11] G. Nikolentzos, A. Tixier, and M. Vazirgiannis, "Message passing attention networks for document understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8544–8551.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.