AML-IT585

Reproducibility

[Re]A Graph-based Relevance Matching Model for Ad-hoc Retrieval

PujaraDreamy,202211005 PandiyanSaran, 202221011 PatelAayushi, 202211025

JhaMukesh,202211070 PatelMisha,202211060

1 Reproducibility Summary

1.1 Scope of Reproducibility

This document presents a series of experiments aimed at replicating the assertions made in the research concerning the Graph-based Relevance Matching Model for Ad-hoc Retrieval, with a specific focus on utilizing the Gated Graph Recurrent Unit (GGRU) and Graph Attention Network (GAT) to implement the model. The main objective of this study is to retrieve pertinent documents based on a given query. In order to comprehensively evaluate the proposed approach, various datasets were employed, and novel variants of Graph Neural Networks (GNN) were explored.

1.2 Methodology

We have devised a software application for constructing our Graph-based Relevance Matching Model utilizing GGRU (Graphical Gated Recurrent Unit) to enhance our system's functionality. Our experimental evaluations were carried out on a RTX2040 12 GB Graphics Processing Unit, providing high-performance computational capabilities for our model's optimization.

1.3 Results

The results are as depicted in the table given below:

.	nDCG@10	AP(rel=2)
GGRU	0.3196	0.2540
GAT	0.36155	0.30139

Our observations indicate that the outcomes attained from GAT (Graph Attention Network) are significantly superior to those generated by GGRU (Graphical Gated Recurrent Unit).

1.4 What was easy

The GGRU code was readily accessible but was implemented in the PyTorch framework, affording a significant savings of time and resources in our implementation process.

1.5 What was Challenging

The existing code suffered from suboptimal performance, manifested through excessive memory consumption, resulting in resource limitations. The observed deficiency necessitated optimization to ensure its compatibility with a 64 GB RAM configuration.

The task of implementing the Graph Attention Network (GAT) model to replace the Graphical Gated Recurrent Unit (GGRU) posed a challenge, demanding rigorous effort and expertise to execute.

2 Introduction

Graph data, which is formed with entities having various types of interactions between them, is typically handled by graph neural networks. Predicting the connections between unrelated items is one of the tasks where GNNs are frequently utilised. Node classification, or classifying the entities in a graph, is another. One such activity that makes use of GNNs for text representation is text classification. Similar to how transformers represent text, they are used to represent text depending on the circumstances. GNNs have the advantage of having lower computing costs than transformers, which is a disadvantage of transformers.

There are two methods for representing text when it comes to representation. One is transductive, where each document is represented based on the words it contains and the entire corpus is thought of as a single graph. However, a large corpus results in high processing costs and scalability problems. Inductive methods are utilised, where each document is represented as a graph for representing the document, to address the scalability issue. The inductive technique is scale-free since it does not require the entire corpus to represent each document. Zhang and co. TexTING performs text classification using a Gated Graph Recurrent unit. Other GNN variations are used in this paper to carry out the same purpose.

3 Scope of Reproducibility

The following are the Scope of reproducibility: • Apply the approach described in TexTING to other GNN variations. • Examine the impact of model complexity and dataset size on the entropy of the GNN model.

This document presents a series of experiments aimed at replicating the assertions made in the research concerning the Graph-based Relevance Matching Model for Ad-hoc Retrieval, with a specific focus on utilizing the Gated Graph Recurrent Unit (GGRU) and Graph Attention Network (GAT) to implement the model. The main objective of this study is to retrieve pertinent documents based on a given query. In order to comprehensively evaluate the proposed approach, various datasets were employed, and novel variants of Graph Neural Networks (GNN) were explored.

4 Methodology

4.1 Dataset Description

The datasets being used are some of the open-source benchmark datasets, including the AG news dataset, the IMDB sentiment analysis dataset, and the DBpedia 2014 ontology dataset1.

TABLE I
DATASET DESCRIPTION

Dataset	Train set	Test set	Classes
IMDB	25,000	25,000	2
AGnews	1,20,000	76,000	4
DBpedia Ontology	5,60,000	70,000	14

We adopt the same approach that Zhang et al. did. Following are the steps: • Producing the feature and adjacency matrices. Learning how to portray text • Using the ReadOut Layer to predict the class Fig. 1(b) provides the methods for using GNN to document categorization.

4.2 Creating the adjacency matrix and feature matrix

Graph G is the designation for the adjacency matrix that is used to express the word cooccurrence. Each unique word in the document has its glove embeddings concatenated to produce a feature matrix. By providing these matrices as input to different GNN variations, richer embeddings of nodes are produced. Adjacency matrix, first The co-occurrence matrix is created by counting each neighbouring word within the window size for a given word (a node in the graph G) and a fixed size of the window. The letter M stands for the adjacency matrix. The word co-occurrence-based adjacency matrix M R, which represents the count of unique terms in the document, is normalised.

$$M_{norm} = \tilde{P}_{ii}^{-\frac{1}{2}} M \tilde{P}_{ii}^{-\frac{1}{2}} \tag{1}$$

where

$$\tilde{P}_{ii} = \sum_j M_{ij} \tag{2}$$

In Inductive GCN we use the $\tilde{M}_{norm}$ as normalised matrix:

$$\tilde{M}_{norm} = \tilde{P}_{ii}^{-\frac{1}{2}} \tilde{M} \tilde{P}_{ii}^{-\frac{1}{2}} \tag{3}$$

Where

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

I_N is added to ensure that each node is connected to itself.

4.3 Learning the text representation

The architecture of the Gated Graph Recurrent Unit, a subtype of the Recurrent Neural Network, was influenced by the Gated Recurrent Unit.

$$l^{t+1} = M_{normalised} F^t W_l \tag{5}$$
$$u^{t+1} = \sigma(W_u^1 l^{t+1} + W_u^2 F^t + bias_u) \tag{6}$$
$$n^{t+1} = \sigma(W_n^1 l^{t+1} + W_u^2 F^t + bias_u) \tag{7}$$
$$\tilde{F}^t = \tanh(W_f^1 l^{t+1} + W_h^1 (n^t \odot F^t) + bias_u) \tag{8}$$
$$F^{t+1} = \tilde{F}^{t+1} \odot u^{t+1} + F^t \odot (1 - u^{t+1}) \tag{9}$$

where Wk, biasu stand for variables that can be altered during training. Update and reset gate representations are indicated by the letters u and n. The letter "t" stands for the quantity of message-passing layers necessary to enhance the node (word) representations. The general architecture of the GGRU is provided in equations 5–9.

5 Experiments

5.1 Evaluation Metric and parameter settings

Because every class in the dataset is equally represented, accuracy is employed as a measurement parameter. The 300-dimensional glove embeddings are used to build the feature matrix. The adjacency matrix is constructed using a conventional window size of 3 for the contextual representation of words. The models were trained using the Adam optimizer using Glorot initialization.

6 Results

6.1 Study on node representation

Graph neural networks enable message transmission between nodes so that information from nodes that are k hops away can be shared to enhance the representation of nodes. A graph containing node n1 and nodes at 1- and 2-hop distances can be seen in Fig. 1(a). The mean and corresponding standard deviation of the cross-entropy values for each of the k values (k 1, 2, 3, 4) are plotted in Figure 2. Although the early epochs' standard deviation values were high, it is obvious that these values dropped over time. This indicates that changing the k values has no effect on the model's ability to learn.

6.2 Entropy of the output

Entropy is a unit of measurement for information content in signals in information theory. By assessing the randomness in the projected output, it is possible to evaluate the effectiveness of the classification model in machine learning. For instance, a classification model with 50% accuracy has a higher entropy (high randomness) than one with 90% accuracy. Prior to creating any output, the classifier model strives to reduce the model's entropy for the best outcomes.// In our experiment, it was found that the Inductive GAT model, which employs more parameters for training and fewer for testing, had higher entropy for a small dataset like IMDB.
other simpler models (Inductive GCN and TextTING). However, as illustrated in Fig. 3, the entropy value decreases initially when a large dataset like the DBPedia ontology is employed. This demonstrates that the entropy loss can converge at the very first epoch when the dataset is large.//

6.3 Performance of the model

As can be seen in Table II, Inductive GAT performs better than the other two models since it employs an attention mechanism. Inductive GAT can lessen the impact of inconsequential relationships between nodes that are ignored by computed lower attention weights, improving the model's output.

7 Results

The results are as depicted in the table given below:

.	nDCG@10	AP(rel=2)
GGRU	0.3196	0.2540
GAT	0.36155	0.30139

Our observations indicate that the outcomes attained from GAT (Graph Attention Network) are significantly superior to those generated by GGRU (Graphical Gated Recurrent Unit).

8 Conclusion and Future Work

It has been discovered that the entropy of the model is affected by the size of the dataset and the model's complexity. Additionally, we demonstrated that Inductive GAT outperformed other models. Future work might involve using global attention to represent each word in the document based on every other word. Additionally, another area for research would be to improve performance over a model of modern transformers.

9 References

1. Pandian, S., Chaurasia, U., Ranjan, S., Saxena, S. (2022). Comparative Study of Inductive Graph Neural Network Models for Text Classification (No. 9135). EasyChair.

2. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. choice, 2640, 660.

3. Zhang, Y., Zhang, J., Cui, Z., Wu, S., Wang, L. (2021, May). A graph-based relevance matching model for ad-hoc retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4688-4696).

4. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.