

Summary of GPT-GNN: Generative Pre-Training of Graph Neural Networks

John Dewey, jcd18c
jcd18c@fsu.edu

1 INTRODUCTION

Graph neural networks (GNNs) demonstrate themselves to be powerful tools for modeling graph-structured data, and pre-training, the process of tuning parameters on one task to improve training on others, has shown tremendous benefits to the performance of many downstream applications in fields such as natural language processing (NLP) and computer vision (CV). However, different tasks performed on the same graph require enough different sets of labeled data for training, and it is usually arduously expensive/infeasible to access a sufficient amount of data, especially with large-scale graphs. For this reason the paper aims to pre-train GNNs for graph mining with the goal of capturing semantic and structural properties of a graph to easily generalize to any downstream tasks within a few fine-tuning steps on graphs of the same domain.

2 EXISTING APPROACHES

2.1 Preliminaries

Preliminary work with GNNs has yielded two basic operators, **Aggregate**(·) and **Extract**(·). The *mean*, *sum*, and *max* functions make up the basic operators of **Aggregate**(·) (but more complex pooling and normalization functions can be used as well), and the **Aggregate**(·) function serves to aggregate neighborhood information of a node. The **Extract**(·) function is a neighbor information extractor, and it uses a node's representation and an edge between two nodes to extract useful information from the source node. There are various GNN architectures proposed under this framework, such as the graph convolutional network (GCN) and GraphSAGE. There are many that also incorporate an attention mechanism into GNNs like the graph attention network (GAT) and heterogeneous graph transformer (HGT), and these attention-based models implement **Extract**(·) by estimating each source node's importance and applying a weighted aggregation based on the estimation. The pre-training framework presented in the paper, GPT-GNN, applies to all these GNN models.

2.2 Existing Pre-training Methods

Previous work proposed pre-training to learn node representations that largely belong to two categories, network/graph embedding and transfer learning setting. Network/graph embeddings can't be used to initialize other models for fine-tuning over other tasks,

so the paper focuses on a transfer learning setting to pre-train a generic GNN to deal with different tasks. Various propositions have been explored for pre-training with unannotated data. For example, Variational Graph Auto-Encoders were proposed to reconstruct graph structures and the previously mentioned GraphSAGE can optimize the Variational Graph Auto-Encoders through unsupervised loss using a similarity metric based on random walks, Graph Infomax can maximize mutual information between node representations from GNNs and a pooled graph representation. These methods showed improvement over pure supervised learning, the learning tasks can be completed by forcing nearby nodes to have similar embeddings which ignores the semantics and structure of a graph.

3 SYSTEM OVERVIEW

3.1 Generative Pre-Training Framework

3.2 Factorizing Attributed Graph Generation

3.3 Efficient Attribute and Edge Generation

3.4 Heterogeneous and Large Graphs

4 EVALUATION

4.1 Experiment Setup

Datasets & tasks. Base GNN model selection. Pre-training baselines.

4.2 Pre-Training and Fine-Tuning Setup

Two setups - time transfer & field transfer. Field transfer more practical, but using both yields best results. Selection of training, validation, and testing sets and limits on labels.

4.3 Results

Discuss paper results.

5 STRENGTHS

The paper rigorously

6 WEAKNESS

The paper uses two different datasets with different downstream tasks, and uses two different evaluation metrics for the sets: MRR for downstream tasks for data from the Open Academic Graph and micro F1-score for downstream tasks performed on data from the Amazon Review Recommendation Dataset. While there may be a reason to use the different metrics, it is not presented in the paper, so it is unknown if they used these metrics for more accurate data or for more impressive results. Furthermore, the team selected training, validation, and testing sets from data in a peculiar manner when setting up pre-training and fine-tuning. Instead of randomly selecting items for each set, they chose nodes from 2014 to 2016 for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Conference'17, July 2017, Washington, DC, USA

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

training, nodes from 2017 for validation, and nodes from 2018 for testing.

7 YOUR SOLUTION

Please provide your solution here. How can you improve this system and address its limitations. XXXXXX