Pretrained Language Models to Solve Graph Tasks in Natural Language

Frederik Wenkel [1 2] **Guy Wolf** [1 2] **Boris Knyazev** [3]

Abstract

Pretrained large language models (LLMs) are powerful learners in a variety of language tasks. We explore if LLMs can learn from graph-structured data when the graphs are described using natural language. We explore data augmentation and pretraining specific to the graph domain and show that LLMs such as GPT-2 and GPT-3 are promising alternatives to graph neural networks.

1. Introduction

Recently, large language models (LLMs) have shown remarkable performance in language tasks (Brown et al., 2020; OpenAI, 2023). For non-language machine learning tasks, domain-specific models have dominated, e.g. convolutional neural networks for images (Wightman et al., 2021) or graph neural networks (GNNs) for graphs (Kipf & Welling, 2016; Veličković et al., 2017; Rampášek et al., 2022). However, Dinh et al. (2022) showed that LLMs, specifically auto-regressive ones such as GPT-3 (Brown et al., 2020), can perform well even on non-language tasks, such as classification and regression on tabular and image data. In such cases, the LLMs are fine-tuned to complete textual prompts, such as ``When we have an image with pixels 0 1 200 .... 0 0, what should be its class?'' [1]. Along this direction, Jablonka et al. (2023) showed that LLMs can be fine-tuned to perform well in the chemical domain. In their case, a string representation (in one of the chemical formats, namely IUPAC names, SMILES, or SELFIES) is provided as a prompt to the LLMs. For instance, the prompt can be ``What is the lipophilicity of COc1cc(N2CCN(C)CC2)c3nc...?'' and the LLMs are fine-tuned to complete the prompt with the molecule's property value (Jablonka et al., 2023). While they showed their approach is competitive with other common methods including GNNs, in particular Wang et al. (2022), they leverage a strong chemical prior by using the IUPAC names, SMILES, or SELFIES representation.

In this work, we aim to understand if LLMs are effective *general* graph learners, when resorting to domain specific representations such as IUPAC names, SMILES, or SELFIES is not possible. Following tabular data prompts in Dinh et al. (2022), we propose to describe graphs via prompts using natural language. To do so, we revisit a simple language developed to describe graphs, the Graph Modelling Language (GML) proposed by Himsolt (1997). For example, a graph with three nodes and three edges can be described as:

```
graph [
   comment "This is a sample graph"
   directed 1
   id 42
   label "Hello, I am a graph"
   node [
      id 1
      label "node 1"
      thisIsASampleAttribute 42
   ]
   node [
      id 2
      label "node 2"
      thisIsASampleAttribute 43
   ]
   node [
      id 3
      label "node 3"
      thisIsASampleAttribute 44
   ]
   edge [
      source 1
      target 2
      label "Edge from node 1 to node 2"
   ]
   edge [
      source 2
      target 3
      label "Edge from node 2 to node 3"
   ]
   edge [
      source 3
      target 1
      label "Edge from node 3 to node 1"
   ]
]
```

[1]Mila - Québec AI Institute [2]Université de Montréal [3]Samsung - SAIT AI Lab, Montreal. Correspondence to: Frederik Wenkel <frederik.wenkel@mila.quebec>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling, Honolulu, Hawaii, USA. 2023.

[1]To train the LLM, it is asked to predict the next token and the loss is computed between the prediction and ground truth token (a digit from 0 to 9 in this example).

Contributions. We demonstrate that pretrained LLMs fine-tuned on GML-based prompts are a promising approach to solve graph learning tasks. Our main contributions are two-fold:

- We show that the GML-based description of graphs can be compressed more than 2 times (in terms of tokens per graph) while only marginally hurting the downstream performance.
- By evaluating GPT-2 and three GPT-3 variants on CYCLES and ZINC (Gómez-Bombarelli et al., 2018; Dwivedi et al., 2020), we show that using stronger LLMs results in better downstream graph performance. While our evaluation shows that using GML-based prompts and LLMs is currently inferior to GNNs, the scaling trend suggests that LLMs may eventually be competitive with GNNs.

We further propose a data augmentation strategy for graph data to improve generalization that is motivated by node permutation invariance principles that are critical for the success of GNN approaches. We also explore pretraining on graph tasks as a potential way to narrow the gap between LLMs and GNNs, but found that such a pretraining only helps in initial fine-tuning iterations, without yielding significant gains at the end of training, which requires further investigation.

2. Related Work

Our work is connected to pure transformers (Kim et al., 2022) that have a weak graph inductive bias and yet are able to compete with GNNs that have a strong graph inductive bias. While pure transformers achieve that by training on large data with millions of samples, we aim to achieve that by leveraging the pretraining power of LLMs. However, we note that LLMs have an even weaker inductive prior than pure transformers because of the absence of explicitly defined node and edge identifiers.

Recent work of Wang et al. (2023) also explored LLMs (GPT-3/4) for solving graph tasks using natural language. They propose a collection of artificial graph problems of increasing complexity to study, e.g., preliminary graph reasoning capacity or the effect of different promting techniques. This is slightly orthogonal to our work as we focus on real-world data like ZINC (Dwivedi et al., 2020) and compare LLM capacity to state-of-the-art GNNs.

Flam-Shepherd & Aspuru-Guzik (2023) showed strong generation abilities for molecular graphs but, similarly to Jablonka et al. (2023), they used chemical string representations which limits the approach to chemical data only. For example, this does not allow for pretraining on synthetic or out-of-domain data, as done in our work.

Table 1. Prompt design approaches with respective number of tokens & mean absolute error (MAE) for ZINC test graphs using GPT-3 (Ada).

Approach	AVG number of tokens per graph	MAE
Full text	963	0.616
Reduced text	400	0.658

3. Method

3.1. Prompt Design

The exact way the prompts are constructed is critical because it affects the ability of the LLM to capture the relationship between the prompt and completion and because it affects the number of tokens and thus the computational cost of fine-tuning. For example, OpenAI's GPT-3 fine-tuning cost and model usage is priced per 1K tokens, e.g., training: \$0.0004 / 1K tokens, usage: \$0.0016 / 1K tokens[2]. Therefore, we consider two approaches to prompting in order to have a good balance between information vs cost and illustrate them using examples from the ZINC molecular dataset (Dwivedi et al., 2020).

Full text. The full text approach closely resembles the Graph Modelling Language. While it appears more informative to a human (than our reduced text approach described next), it is lengthy and is constructed as following.

Prompt:

```
What is the constrained solubility
(penalized logP) of a molecular graph
with 9 nodes and 16 edges described
based on Graph Modelling Language as:
node [ id=0, label=C ], node [ id=1,
label=C ], node [ id=2, label=O ],
node [ id=3, label=C ], node [ id=4,
label=O ], node [ id=5, label=CH1 ],
node [ id=6, label=Cl ], node [ id=7,
label=C ], node [ id=8, label=C ], edge
[ source=0, target=1, label=1 ], edge
[ source=1, target=0, label=1 ], edge
[ source=1, target=2, label=1 ], ...  ,
edge [ source=8, target=7, label=1 ]?.
```

Completion:

```
The logP score is -0.2089.
```

Reduced text. To reduce the number of tokens without losing essential graph information, we consider constructing our prompts as following.

Prompt:

```
graph with 9 nodes:  0 C , 1 C , 2 O ,
```

[2]https://openai.com/pricing

```
3 C , 4 O , 5 CH1 , 6 Cl , 7 C , 8 C;
edges:  0 1 single , 1 0 single , 1 2
single , ...  , 8 7 single.
```

Completion:

```
score:  -0.2089.
```

Further study of different promt designs and compression strategies can be found in Section **??** in the appendix.

3.2. Data Augmentation

As demonstrated in Dinh et al. (2022), it is beneficial to employ data augmentation strategies for improved generalization of LLMs. We follow this approach as well by permuting the node identifiers (numbers 0 to 8 in the prompt example above) and associated edge source and targets passed to the LLMs. This serves two objectives: (i) avoiding the models to "fingerprint" individual molecules, and (ii) encouraging the LLMs to learn node(identifier)-permutation invariant representations, which is a fundamental practice in graph learning (Wu et al., 2020) and satisfied for most GNNs by design.

4. Experiments

Setup. We mainly use GPT-2 in our experiments (Radford et al., 2019) as it is computationally feasible and simple to fine-tune. We follow standard train, validation and test splits of the ZINC dataset with 10k train, 1k validation and 1k test graphs (Dwivedi et al., 2020). In the case of the CYCLES dataset, we use 9k train, 1k validation and 10k test graphs.

We experiment with different training regimes: **GPT-2** and **GPT-3**, are the iterations of the GPT model, respectively, pretrained on language data. We further add the specifier **+aug** if the proposed augmentation strategy was employed, and **+cyc** if additional pretraining was conducted on the CYCLES dataset (Dwivedi et al., 2020). For completeness, we also include a vanilla GPT2 model, **GPT-2-scratch**, that has not been pretrained at all.

For GPT-3 fine-tuning we use the official instruction[3] and fine-tune three variants (ada, babbage and curie).

Results. In Table 2, we report results on the CYCLES and ZINC dataset (Dwivedi et al., 2020) for a selection of LLM architectures and compare them to a selection of popular GNN models.

On the ZINC dataset, we observe that subsequent iterations of the GPT model perform increasingly well, with GPT-3 (Curie) approaching specialized graph learning approaches such as GIN (Xu et al., 2019). We also found that training a larger GPT-3 variant (Babbage) on reduced prompts lead to the same results (0.616) as training a smaller GPT-3 variant (Ada) on full text prompts, while being cheaper ($9.78 vs $15.59). Thus, further reducing the prompt size while increasing the model capacity may be a feasible approach to improve results.

[3]https://platform.openai.com/docs/guides/fine-tuning

Table 2. Main results on CYCLES (accuracy; higher is better) and ZINC (MAE; lower is better). See more results in Table **??**.

MODEL	CYCLES	ZINC
GPT-2-SCRATCH	0.643	0.858
GPT-2	0.649	0.764
GPT-2+AUG	0.649	0.723
GPT-2+AUG+CYC	N/A	0.763
GPT-3 (ADA)	-	0.658
GPT-3 (BABBAGE)	-	0.616
GPT-3 (CURIE)	-	0.584
NON LM BASELINES:		
MLP	-	0.706
GCN (KIPF & WELLING, 2016)	-	0.459
GIN (XU ET AL., 2019)	0.861	0.526
GAT (VELIČKOVIĆ ET AL., 2017)	-	0.384
GPS (RAMPÁŠEK ET AL., 2022)	-	0.070

We further investigate the merit of several strategies to improve LLM performance in the case of GPT-2. We first observe that language pretraining (**GPT-2**) clearly performs better compared to training the model from scratch (**GPT-2-scratch**), indicating that language-pretrained LLMs are good candidates to be fine-tuned for graph learning tasks.

Second, we observe that the proposed data augmentation strategy (Section 3.2) further improves performance noticeable, suggesting that it is beneficial to encourage LLMs to "understand" the concept of permutation invariance through training.

We also experimented with additional pretraining on graph data (**GPT-2+aug+cyc**) using the CYCLES dataset (Dwivedi et al., 2020). We expect this task to be beneficial for pretraining for ZINC, as the target (constrained solubility), apart from other values, depends on the number of cycles of minimal length of six atoms. While this specific pretraining had a negative effect on the final test performance (Table 2), the model fine-tuning initially progressed faster (e.g., reaching lower training loss faster) as seen in Figure 1. As in Dwivedi et al. (2020), we only experimented with graphs of size 56 nodes and cycles of length six here and hypothesize that refining the graph-pretraining task (e.g., by including cycles of various lengths) could further improve the performance on ZINC.

For the CYCLES dataset, we only observe a slight improvement of the language-pretrained model **GPT-2** over **GPT-2-scratch** in terms of final test accuracy. However, the language-pretrained model training converges faster.

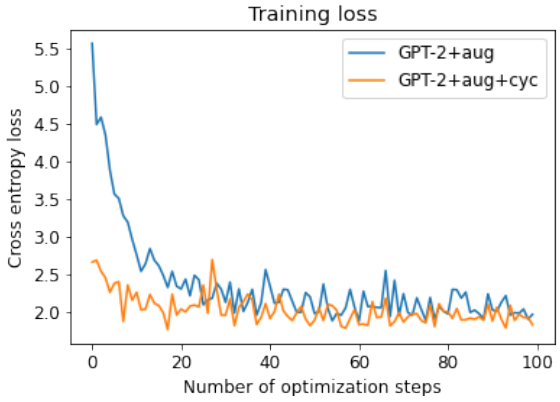

Figure 1. Training loss on ZINC dataset for first 100 optimization steps for **GPT-2+aug+cyc** and **GPT-2+aug**. Pretraining on CYCLES leads to lower initial training loss, suggesting the pretraining is informative for the ZINC task.

5. Conclusion

Despite relatively low performance of GPT models, we believe that the approach of fine-tuning LLMs using natural language has a lot of promise. In particular, this approach allows for leveraging vast knowledge of LLMs. The presented results show a clear scaling trend on the ZINC dataset when moving from older to newer GPT models, promising further advances when moving to even newer versions. In addition, it allows for pretraining on diverse graph tasks alleviating the node and edge feature mismatch issue. However, more research is required to develop the best practice of such a pretraining. Another interesting direction is further prompt compression, exploring alternative "language" to describe graphs and allowing to tackle tasks on datasets with larger graphs.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020. 1

Dinh, T., Zeng, Y., Zhang, R., Lin, Z., Gira, M., Rajput, S., Sohn, J.-y., Papailiopoulos, D., and Lee, K. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022. 1, 3

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. 2020. 2, 3

Flam-Shepherd, D. and Aspuru-Guzik, A. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files. *arXiv preprint arXiv:2305.05708*, 2023. 2

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018. 2

Himsolt, M. Gml: A portable graph file format. Technical report, Technical report, Universitat Passau, 1997. 1

Jablonka, K. M., Schwaller, P., Ortega-Guerrero, A., and Smit, B. Is gpt-3 all you need for low-data discovery in chemistry? 2023. 1, 2

Kim, J., Nguyen, D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems*, 35: 14582–14595, 2022. 2

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 3

OpenAI. Gpt-4 technical report, 2023. 1

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022. 1, 3

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 1, 3

Wang, H., Feng, S., He, T., Tan, Z., Han, X., and Tsvetkov, Y. Can language models solve graph problems in natural language? *arXiv preprint arXiv:2305.10037*, 2023. 2

Wang, Y., Wang, J., Cao, Z., and Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022. 1

Wightman, R., Touvron, H., and Jégou, H. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 1

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. 3

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 3