Paper Discussion & Teammate Finding Ⅱ

Jiaxuan You
Assistant Professor at UIUC CDS

CS598: Deep Learning with Graphs, 2024 Fall
https://ulab-uiuc.github.io/CS598/

Paper Discussion & Teammate Finding

Recommended Papers

Overview

- Most popular papers in analysis & recommendations
  - Can Language Models Solve Graph Problems in Natural Language? [link, NeurIPS 2023]
  - Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs [link, Findings of ACL 2024]
  - G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering [link, arXiv]
  - LLaGA: Large Language and Graph Assistant [link, ICML 2024]
  - PRODIGY: Enabling In-context Learning Over Graphs [link, NeurIPS 2023]

Recommended Papers (1)

- ***Can Language Models Solve Graph Problems in Natural Language?***

  Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, Yulia Tsvetkov

Can Language Models Solve Graph Problems in Natural Language?

Heng Wang*[1], Shangbin Feng*[2], Tianxing He[2], Zhaoxuan Tan[3], Xiaochuang Han[2], Yulia Tsvetkov[2]
[1]Xi'an Jiaotong University [2]University of Washington [3]University of Notre Dame
wh2213210554@stu.xjtu.edu.cn, shangbin@cs.washington.edu

arXiv:2305.10037v3 [cs.CL] 6 Jan 2024

Abstract

Large language models (LLMs) are increasingly adopted for a variety of tasks with *implicit graphical structures*, such as planning in robotics, multi-hop question answering or knowledge probing, structured commonsense reasoning, and more. While LLMs have advanced the state-of-the-art on these tasks with structure implications, whether LLMs could explicitly process textual descriptions of graphs and structures, map them to grounded conceptual spaces, and perform structured operations remains underexplored. To this end, we propose NLGraph (Natural Language Graph), a comprehensive benchmark of graph-based problem solving designed in natural language. NLGraph contains 29,370 problems, covering eight graph reasoning tasks with varying complexity from simple tasks such as connectivity and shortest path up to complex problems such as maximum flow and simulating graph neural networks. We evaluate LLMs (GPT-3/4) with various prompting approaches on the NLGraph benchmark and find that 1) language models *do* demonstrate preliminary graph reasoning abilities, 2) the benefit of advanced prompting and in-context learning diminishes on more complex graph problems, while 3) LLMs are also (un)surprisingly brittle in the face of spurious correlations in graph and problem settings. We then propose Build-a-Graph Prompting and Algorithmic Prompting, two instruction-based approaches to enhance LLMs in solving natural language graph problems. Build-a-Graph and Algorithmic prompting improve the performance of LLMs on NLGraph by 3.07% to 16.85% across multiple tasks and settings, while how to solve the most complicated graph reasoning tasks in our setup with language models remains an open research question. The NLGraph benchmark and evaluation code are available at https://github.com/Arthur-Heng/NLGraph

1 Introduction

Originally designed for textual data, large language models (LLMs) are increasingly leveraged for tasks beyond language processing. In robotics and planning, LLMs are adopted to guide agents through structured environments [Huang et al., 2022, Andreas, 2022]. In theory-of-mind reasoning, LLMs are required to maintain and update local and global graphs that reflect the beliefs of different characters [Adhikari et al., 2020, Ammanabrolu and Riedl, 2021]. In structured commonsense reasoning, LLMs are expected to generate graph-based action plans to achieve objectives with diversified prerequisites [Tandon et al., 2019, Madaan et al., 2022]. In multi-hop question answering, LLMs implicitly find connections and paths among a vast network of entities and concepts [Creswell et al., 2023]. Together these works demonstrate that LLMs are widely adopted for tasks with *implicit graphical structures* while achieving preliminary success. However, one underlying yet crucial question remains underexplored: *Can LLMs reason with graphs?* More concretely, *are LLMs capable of mapping textual descriptions of graphs and structures to grounded conceptual spaces and solving*

*equal contribution

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Recommended Papers (2)

- ***Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs***

  Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, Jiawei Han

Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs

Bowen Jin[1], Chulin Xie[1], Jiawei Zhang[1], Kashob Kumar Roy[1], Yu Zhang[1] Zheng Li[2], Ruirui Li[2], Xianfeng Tang[2], Suhang Wang[3], Yu Meng[4], Jiawei Han[1]
[1]University of Illinois at Urbana-Champaign, [2]Amazon
[3]Pennsylvania State University, [4]University of Virginia
bowenj4@illinois.edu

Abstract

Large language models (LLMs), while exhibiting exceptional performance, suffer from hallucinations, especially on knowledge-intensive tasks. Existing works propose to augment LLMs with individual text units retrieved from external knowledge corpora to alleviate the issue. However, in many domains, texts are interconnected (*e.g.*, academic papers in a bibliographic graph are linked by citations and co-authorships) which form a (text-attributed) graph. The knowledge in such graphs is encoded not only in single texts/nodes but also in their associated connections. To facilitate the research of augmenting LLMs with graphs, we manually construct a **G**raph **R**easoning Benchmark dataset called GRBENCH, containing 1,740 questions that can be answered with the knowledge from 10 domain graphs. Then, we propose a simple and effective framework called Graph Chain-of-thought (GRAPH-COT) to augment LLMs with graphs by encouraging LLMs to reason on the graph iteratively. Each GRAPH-COT iteration consists of three sub-steps: LLM reasoning, LLM-graph interaction, and graph execution. We conduct systematic experiments with three LLM backbones on GRBENCH, where GRAPH-COT outperforms the baselines consistently. The code is available at https://github.com/PeterGriffinJin/Graph-CoT.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023; Jiang et al., 2024) have demonstrated their exceptional language understanding and text generation capability in real-world scenarios (Zhao et al., 2023). However, LLMs suffer from hallucination problems and sometimes tend to generate content that appears plausible but is ungrounded (Tonmoy et al., 2024). This is because they memorize world knowledge parametrically and fail to refer to concrete knowledge sources (Zhang et al., 2023b). To alleviate the hallucination issues, existing works propose to augment LLMs with external text corpora as knowledge sources (Shuster et al., 2021; Wu et al., 2023) and treat every single document as a knowledge unit. Retrieval augmentation (RAG) (Lewis et al., 2020; Gao et al., 2023) is then proposed to enable LLMs to interact with external knowledge sources, where relevant texts are retrieved and serve as contexts to improve the factuality of LLMs (shown in Figure 1 (a)). However, retrieval augmentation assumes that knowledge is well represented in individual text units and ignores the correlations among multiple text units.

In real-world scenarios, text units are generally interconnected, forming a (text-attributed) graph. The knowledge of such graphs is reflected not only in the form of texts but also in the structure of their connections. For example, academic papers in a bibliographic graph are linked by citation links (Wang et al., 2020). We can trace the source of a research direction (Bai et al., 2019) by traversing such a graph. Cases and opinions in a legal graph are interconnected by reference edges (Sadeghian et al., 2018). We can verify the judgment for a case by looking up its citations on such a graph (Chen et al., 2019).

Although widely used for text corpora as external knowledge sources, retrieval-augmentation cannot be readily used to augment LLMs with graphs for two reasons: 1) *Structure Context*: Retrieval augmentation can find individual nodes/texts from the graphs which can serve as context to augment the LLMs. However, knowledge on the graph also lies in the structure which can not be captured by single nodes/texts. 2) *Graph Size Explosion*: Although it is feasible to convert local subgraph structures into text descriptions as the input contexts to LLMs, the size of the local subgraph increases exponentially as the hop number increases, resulting in an excessively long context sequence. This could cause LLMs to be lost in the middle (Liu

Recommended Papers (3)

- ***G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering***

  Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, Bryan Hooi

arXiv:2402.07630v3 [cs.LG] 27 May 2024

G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering

Xiaoxin He[1] Yijun Tian[2] Yifei Sun[1] Nitesh V. Chawla[2] Thomas Laurent[3] Yann LeCun[4,5] Xavier Bresson[1] Bryan Hooi[1]

{xiaoxin, yifeisun, xaviercs, bhooi}@comp.nus.edu.sg
{yijun.tian, nchawla}@nd.edu, tlaurent@lmu.edu, yann@cs.nyu.edu

[1]National University of Singapore [2]University of Notre Dame [3]Loyola Marymount University [4]New York University [5]Meta AI

Abstract

Given a graph with textual attributes, we enable users to 'chat with their graph': that is, to ask questions about the graph using a conversational interface. In response to a user's questions, our method provides textual replies and highlights the relevant parts of the graph. While existing works integrate large language models (LLMs) and graph neural networks (GNNs) in various ways, they mostly focus on either conventional graph tasks (such as node, edge, and graph classification), or on answering simple graph queries on small or synthetic graphs. In contrast, we develop a flexible question-answering framework targeting real-world textual graphs, applicable to multiple applications including scene graph understanding, common sense reasoning, and knowledge graph reasoning. Toward this goal, we first develop a Graph Question Answering (GraphQA) benchmark with data collected from different tasks. Then, we propose our *G-Retriever* method, introducing the first retrieval-augmented generation (RAG) approach for general textual graphs, which can be fine-tuned to enhance graph understanding via soft prompting. To resist hallucination and to allow for textual graphs that greatly exceed the LLM's context window size, *G-Retriever* performs RAG over a graph by formulating this task as a Prize-Collecting Steiner Tree optimization problem. Empirical evaluations show that our method outperforms baselines on textual graph tasks from multiple domains, scales well with larger graph sizes, and mitigates hallucination.[1]

1 Introduction

Graphs and Large Language Models (LLMs). The advent of LLMs has significantly shaped the artificial intelligence landscape. As these models are applied to increasingly diverse tasks, their ability to process complex structured data will be increasingly vital. In particular, in our interconnected world, a significant portion of real-world data inherently possesses a graph structure, such as the Web, e-commerce, recommendation systems, knowledge graphs, and many others. Moreover, many of these involve graphs with textual attributes (*i.e., textual graphs*), making them well-suited for LLM-centric methods. This has spurred interest in combining graph-based technologies, particularly graph neural networks (GNNs), with LLMs to enhance their reasoning on graphs [44, 15, 24].

The Present Work: Enabling 'Chat With Your Graph'. While existing works integrate LLMs and GNNs in various ways, they mostly focus on conventional graph tasks such as node, edge and graph

[1]Our codes and datasets are available at: `https://github.com/XiaoxinHe/G-Retriever`

Recommended Papers (4)

- ***LLaGA: Large Language and Graph Assistant***

  Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, Zhangyang Wang

LLaGA: Large Language and Graph Assistant

Runjin Chen [1] Tong Zhao [2] Ajay Jaiswal [1] Neil Shah [2] Zhangyang Wang [1]

Abstract

Graph Neural Networks (GNNs) have empowered the advance in graph-structured data analysis. Recently, the rise of Large Language Models (LLMs) like GPT-4 has heralded a new era in deep learning. However, their application to graph data poses distinct challenges due to the inherent difficulty of translating graph structures to language. To this end, we introduce the **L**arge **L**anguage **a**nd **G**raph **A**ssistant (**LLaGA**), an innovative model that effectively integrates LLM capabilities to handle the complexities of graph-structured data. LLaGA retains the general-purpose nature of LLMs while adapting graph data into a format compatible with LLM input. LLaGA achieves this by reorganizing graph nodes to structure-aware sequences and then mapping these into the token embedding space through a versatile projector. LLaGA excels in versatility, generalizability and interpretability, allowing it to perform consistently well across different datasets and tasks, extend its ability to unseen datasets or tasks, and provide explanations for graphs. Our extensive experiments across popular graph benchmarks show that LLaGA delivers outstanding performance across four datasets and three tasks using one single model, surpassing state-of-the-art graph models in both supervised and zero-shot scenarios. Our code is available at `https://github.com/VITA-Group/LLaGA`

1. Introduction

Graphs are omnipresent, representing a myriad of real-world data from social networks, biological networks and recommendation systems, etc. Graph neural networks (GNNs) (Kipf & Welling, 2017; Defferrard et al., 2016; Veličković et al., 2017), embedded with message passing and aggregation techniques, are powerful algorithmic tools on handling complex graph structures. Nonetheless, a critical limitation of GNNs is their weak multi-task handling capability. Typically trained on a single task, GNNs struggle to maintain performance when applied to multiple tasks. Self-supervised learning (Jin et al., 2021; Ju et al., 2023) may offer some improvement, but they still require task-specific heads or tuning for downstream tasks.

Recently, the advent of LLMs having massive context-aware knowledge and semantic comprehension capabilities (*e.g.*, LLaMa (Touvron et al., 2023), GPTs (Achiam et al., 2023), Claude (Perez et al., 2022)) marks a significant advancement in AI research. A key advantage of LLMs is their ability to solve various tasks with a single model, showcasing strong language skills and the capacity to explain provided answers. These models have demonstrated remarkable proficiency not only in language-related tasks but also in understanding and generating visual content (Liu et al., 2023; Wang et al., 2023). However, direct application of such models presents challenges when it comes to graph-structured data, which inherently contains rich relational and structural information. Hence, researchers (Fatemi et al., 2023; Chen et al., 2023a) explored ways to translate graph structures into natural language suitable for consumption by language models. Yet, describing graphs in plain texts tends to be verbose and fails to directly represent the intrinsic characteristics of graphs, often leading to repetitive and unintuitive descriptions of nodes and edge relationships. Consequently, LLMs would perform poorly on basic graph tasks without specific adaptations (Chen et al., 2023a). Subsequently, InstructGLM (Ye et al., 2023) describes graphs in language and attempts to enhance LLMs' graph-task performance by task-specific fine-tuning. However, this specialization constrains the model's versatility, potentially limiting its effectiveness in other graph tasks or non-graph-related domains. More recently, GraphGPT (Tang et al., 2023) has combined text descriptions with a self-supervised graph transformer to incorporate graph data into large language models (LLMs). However, the pre-trained graph transformer might not distill all relevant structural information for specific downstream tasks, leading to less satisfactory performances. Motivated by these issues, this work poses an important **question**: *How to develop a framework that effectively encodes structural information for graphs across various tasks and domains, enabling its comprehension by LLMs, while maintaining*

[1] The University of Texas at Austin [2] Snap Inc. Correspondence to: Zhangyang Wang <atlaswang@utexas.edu>, Runjin Chen <chenrunjin@utexas.edu>.

Preprint.

1

Recommended Papers (5)

- ***PRODIGY: Enabling In-context Learning Over Graphs***

  Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy Liang, Jure Leskovec

arXiv:2305.12600v1 [cs.LG] 21 May 2023

PRODIGY: Enabling In-context Learning Over Graphs

Qian Huang[1*] qhwang@cs.stanford.edu
Hongyu Ren[1*] hyren@cs.stanford.edu
Peng Chen[1] pengc@stanford.edu
Gregor Kržmanc[2] gregor.krzmanc@ijs.si
Daniel Zeng[1] dzeng@cs.stanford.edu
Percy Liang[1] pliang@cs.stanford.edu
Jure Leskovec[1] jure@cs.stanford.edu

[1] Stanford University
[2] University of Ljubljana

Abstract

In-context learning is the ability of a pretrained model to adapt to novel and diverse downstream tasks by conditioning on prompt examples, without optimizing any parameters. While large language models have demonstrated this ability, how in-context learning could be performed over graphs is unexplored. In this paper, we develop **Pr**etraining **O**ver **D**iverse **I**n-Context **G**raph S**y**stems (PRODIGY), the first pretraining framework that enables in-context learning over graphs. The key idea of our framework is to formulate in-context learning over graphs with a novel *prompt graph* representation, which connects prompt examples and queries. We then propose a graph neural network architecture over the prompt graph and a corresponding family of in-context pretraining objectives. With PRODIGY, the pre-trained model can directly perform novel downstream classification tasks on unseen graphs via in-context learning. We provide empirical evidence of the effectiveness of our framework by showcasing its strong in-context learning performance on tasks involving citation networks and knowledge graphs. Our approach outperforms the in-context learning accuracy of contrastive pretraining baselines with hard-coded adaptation by 18% on average across all setups. Moreover, it also outperforms standard finetuning with limited data by 33% on average with in-context learning.

1 Introduction

In-context learning is a novel and one of the most intriguing capabilities of language models [1]. It refers to the capability of a pretrained model to perform novel and diverse tasks directly at the prediction time when prompted with just a few examples, without the need to update the model weights. For example, a person may describe the new task (*e.g.*, question answering, machine translation, or code generation) using natural language and demonstrate it to the language model with several prompt examples. The language model then directly without any model training or finetunning performs the task.

However, how to enable in-context learning for diverse graph machine learning tasks, such as identifying misinformation spreader in social networks [14] and product suggestions across online e-commerce websites [21], still remain unexplored and challenging. An in-context learner for graphs should be able to solve novel tasks on novel graphs. For example, give music product

*indicates equal contribution.

Preprint. Under review.

Paper Discussion & Teammate Finding

Idea Proposal & Feedback

Proposal Task

- Guidelines
  - Answer the 5 questions for your research project
    - **Be specific**. Equivalent to write an Introduction for your project paper.
    - Include citations discussions to relevant literature
    - Describe your expected results, but no need to run any experiment.
    - Will be discussed with your classmates in another discussion session on Oct 9
- Submission
  - Submitted a PDF file to Canvas for each group (group assignment finalized by next Monday). Suggested length: 1.5-2 pages excluding references.
  - **Submission deadline: Oct 6 11:59 PM, CT**
  - Accounts for **10%** of final grade (Idea & discussion category)

Schedule and Process

- Groups pairing
  - Find your project teammates, 3 people each by default
  - Sit with another group to for discussion, 6 people each by default, see the Google Sheet for potential mates [link, also in Slack]
- Project brainstorming via 4 out of 5 questions
  - Present your ideas based on your reading and analysis
  - Comment on each other's ideas to provide feedback
- Feedback seeking
  - Ask the other group sitting together with you for feedback
  - Reach out to the instructor for advice

5 Questions for Discussion

- **What is the problem?** (12:50 – 13:05)
  - Brainstorm at least 3 ideas
- **Why is it interesting and important?** (13:20 – 13:30)
- **Why is it hard?** (E.g., why do naive approaches fail?) (13:30 – 13:35)
- **Why hasn't it been solved before?** (Or, what's wrong with previous proposed solutions? How does mine differ?) (13:35 – 13:45)

- What are the key components of my approach and results? Also include any specific limitations. (Optional)