LLMs for Graphs

HTET ARKAR

Junior | Undergraduate

School of Computer Science and Engineering

Chung-Ang University

21-Jan-25

A Survey of Large Language Models for Graphs

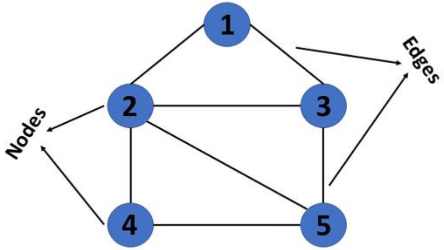
Xubin Ren[1], Jiabin Tang[1], Dawei Yin[2], Nitesh Chawla[3], Chao Huang[1]
[1]University of Hong Kong, [2]Baidu Inc., [3]University of Notre Dame

KDD '24

Contents

Introduction

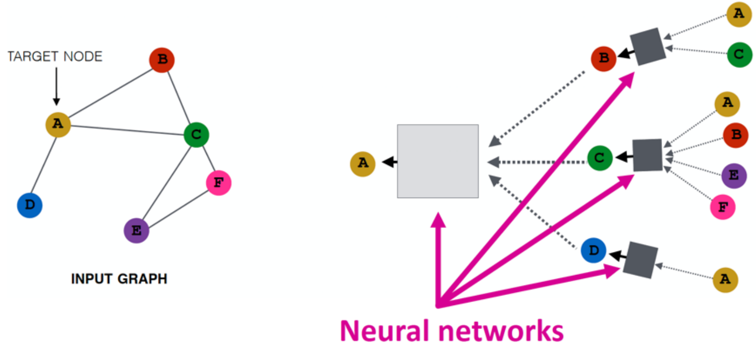
- **Graphs**
  - **Structured data form**
  - Comprising **nodes** and **edges** that signify relationships
  - Essential for real-world connections across various domains
    - Social Networks
    - Molecular Graphs
    - Recommender Systems
    - Academic Networks



Graphs

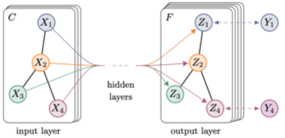
Graph Neural Network (GNNs)

- Passing and aggregating information across nodes
- Iteratively refining node features through supervised learning
- Remarkable results in capturing structural nuances
- Enhance model accuracy

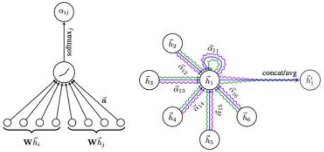


Graphs

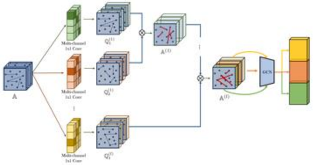
- **Graph Convolutional Networks (GCNs)**
  - Effective in propagation embedding across nodes



- **Graph Attention Networks (GATs)**
  - Leverage attention mechanisms
  - Perform precise aggregation of node features

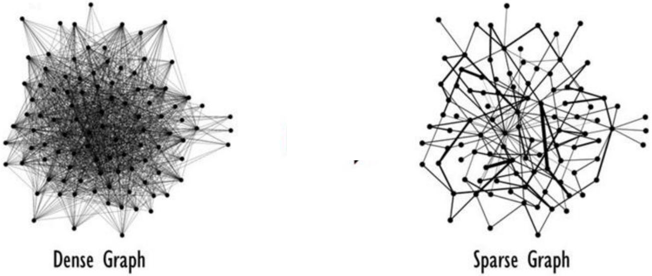


- **Graph Transformers**
  - Employe self-attention and positional encoding
  - Capture global signals among graph + Improving expressiveness



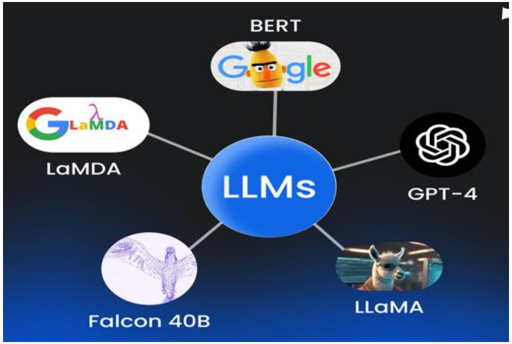
Graphs

- **Challenges**
  - Data Sparsity
    - Particularly in scenarios where graph structure is incomplete or noisy
  - Generalization ability
    - New graphs or unseen nodes
- **Need for more robust and adaptive models**



Introduction (Cont'd)

- **Large Language Models**
  - Great generalization abilities for unseen tasks
  - Powerful tools in various research fields
    - Natural Language Processing (NLP)
    - Computer Vision
    - Informational Retrieval



Graph Learning Communities

LLMs + GNNs ⟶ Powerful new waves of Methods

- Prompting Investigations into the potential of LLMs
- To enhance performance on graph-related tasks
- Not only improving task performance
- But also demonstrating impressive zero-shot generalization capabilities

Definitions

- **Three Keywords**
  - Graph-Structured Data
  - Graph Neural Networks (GNNs)
  - Large Language Models (LLMs)

Graph-Structured Data

Graph

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- A non-linear data structure
- Edge (u, v) – directed : orientation, undirected : no orientation

Text-Attributed Graph (TAG)

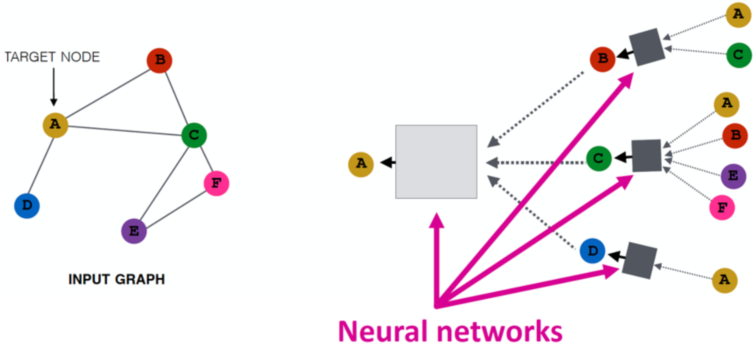
- A node is associated with a sequential text feature ($i.e.$, sentence)
- $\mathcal{G}_S = (\mathcal{V}, \mathcal{E}, \mathcal{T})$
- $\mathcal{T}$: a set of text features

Definitions

Graph Neural Networks (GNNs)

- Deep learning architectures for graph-structured data
- Aggregate information from neighboring nodes
- Update node embeddings by stacking L layers
- Final node embeddings can be used for downstream tasks
  - Node classification and link prediction

$$\mathbf{h}_v^{(l+1)} = \psi(\phi(\{\mathbf{h}_{v'}^{(l)} : v' \in \mathcal{N}(v)\}), \mathbf{h}_v^{(l)})$$



Large Language Models (LLMs)

- A statistical model that estimates the probability distribution of words for a given sentence
- Superior performance in solving a wide range of natural language tasks
- Recent LLMs are built with transformer blocks
  - Use a query-key-value (QKV)-based attention mechanism
  - Aggregate information in the sequence of tokens

LLMs for Graphs

- GNNs as Prefix
- LLMs as Prefix
- LLMs-Graphs Integration
- LLMs-only

GNNs as Prefix

- Application of GNNs: structural encoders
- GNN generally play the role of a tokenizer
- Graph data → graph token sequence (rich in structural information)
- Then input into LLMs to align with natural language
- Divided into two categories:
  1) Node-level Tokenization
  2) Graph-level Tokenization

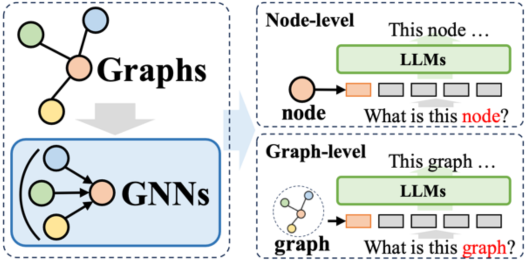


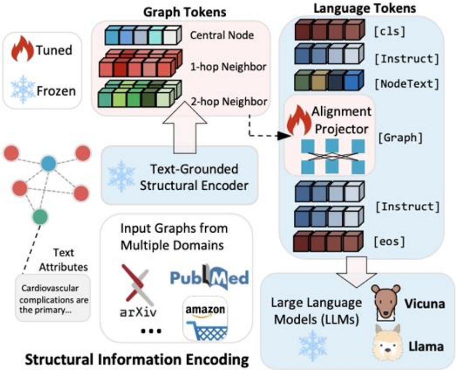
Figure 2: GNNs as Prefix.

GNNs as Prefix

Node-level Tokenization

- For some downstream tasks in Graph Learning, models need to
  - Model the fine-grained structural information at node level
  - Distinguish the semantic differences between different nodes

- GraphGPT
  - Initially align the **graph encoder** with natural language semantics through text-graph grounding
  - Then combine the trained graph encoder with the LLM using a projector



Node-level Tokenization

- For some downstream tasks in Graph Learning, models need to
  - Model the fine-grained structural information at node level
  - Distinguish the semantic differences between different nodes

- GraphGPT
  - Initially align the **graph encoder** with natural language semantics through text-graph grounding
  - Then combine the trained graph encoder with the LLM using a projector

Graph Matching
Graph Information: <graph>: **Central Node:** 68442, **Edge index:** [[…src node…],[…dst node…]], **Node list:** […]
Human Question: Given a sequence of graph tokens **<graph>** that constitute a subgraph of a citation graph, …. Here is a list of paper titles: 1. … 2. …, please reorder the list of papers according to the order of graph tokens.
GraphGPT Response: Based on the given graph tokens and the list of paper titles, we obtain the matching of graph tokens and papers: Graph token 1 corresponds to smt based induction methods for timed systems. Graph token 2 corresponds to …

Node Classification
Graph Information: <graph>: **Central Node:** 2, **Edge index:** [[…src node…],[…dst node…]], **Node list:** […]
Human Question: Given a citation graph: **<graph>** where the 0th node is the target paper, with the following information: Abstract: … Title: … Question: Which arXiv CS sub-category does this paper belong to? …
GraphGPT Response: cs.IT, cs.LG, cs.SP, cs.CV, cs.NA. The paper discusses the Restricted Isometry …. So, it is likely to belong to cs.IT…

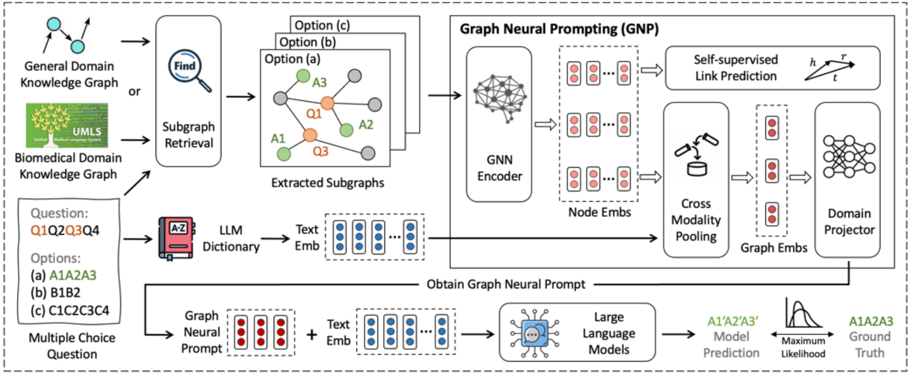
Link Prediction
Graph Information: <graph>: **Central Node 1:** 8471, **Edge index 1:** [[…src node…],[…dst node…]], **Node list 1:** […]
<graph>: **Central Node 2:** 19368, **Edge index 2:** [[…src node…],[…dst node…]], **Node list 2:** […]
Human Question: Given a sequence of graph tokens: **<graph>** that constitute a subgraph of a citation graph, …. Abstract: … Titile: … and the other sequence of graph tokens: **<graph>**, … Abstract: … Title: …, are these two central nodes connected? Give me an answer of "yes" or "no".
GraphGPT Response: Yes, they are connected. Based on the first paper, …. And the second paper proposes ….

Figure 4: Our instruction designs for graph matching task (upper), node classification (middle) and link prediction (lower).

Graph-level Tokenization

- To adapt to other graph-level tasks, models need to
  - Be able to extract graph information from node representations
  - To obtain high-level graph semantic tokens

- GNP
  - Employ cross-modality pooling to integrate the node representations encoded by the graph encoder with the natural language tokens
  - Resulting in a unified graph representation

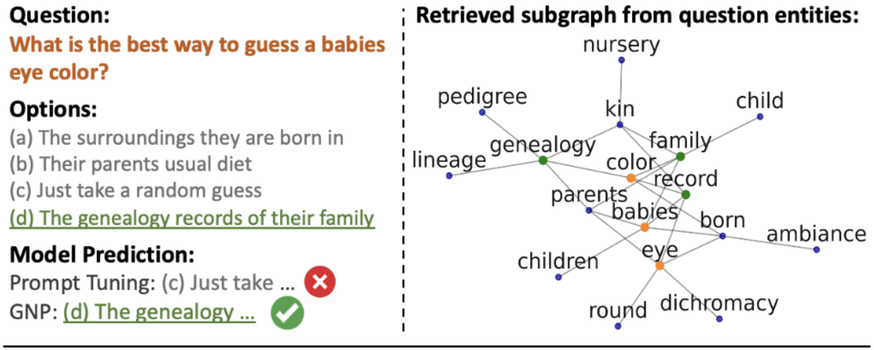


GNNs as Prefix

Graph-level Tokenization

- To adapt to other graph-level tasks, models need to
  - Be able to extract graph information from node representations
  - To obtain high-level graph semantic tokens

- GNP
  - This representation is aligned with the instructions through the LLM
  - To apply in QA tasks
  - To demonstrate superiority in commonsense and biomedical reasoning tasks



Results
&
Challenges

(GNNs as Prefix)

- Unprecedented generalization
  (zero-shot capability)

- Effective for non-text-attributed graphs

LLMs as Prefix

- Leverage the information produced by LLMs to improve training of GNNs
- This information includes:
  - Textual content
  - Labels, or
  - Embeddings derived from LLMs
- Derived into two categories:
  1) Embeddings from LLMs for GNNs
  2) Labels from LLMs for GNNs

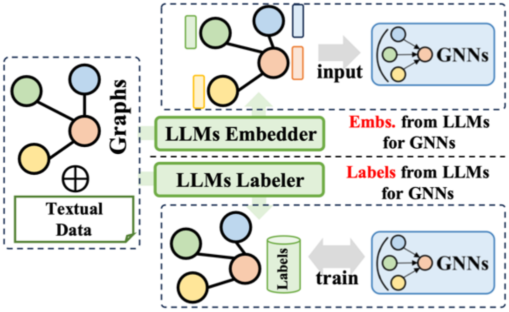
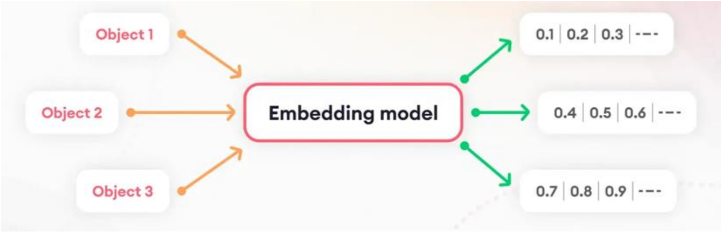


Figure 3: LLMs as Prefix.

Embeddings from LLMs for GNNs

- Initial node embeddings are diverse across different domains
  - ID-based embeddings in RecSys, or bag-of-words embeddings in citation networks
- Poor quality of embeddings can result in suboptimal performance of GNN
- Lack of a universal design for node embedders
- Challenging to address the generalization capability of GNNs
  - In unseen tasks with different node sets

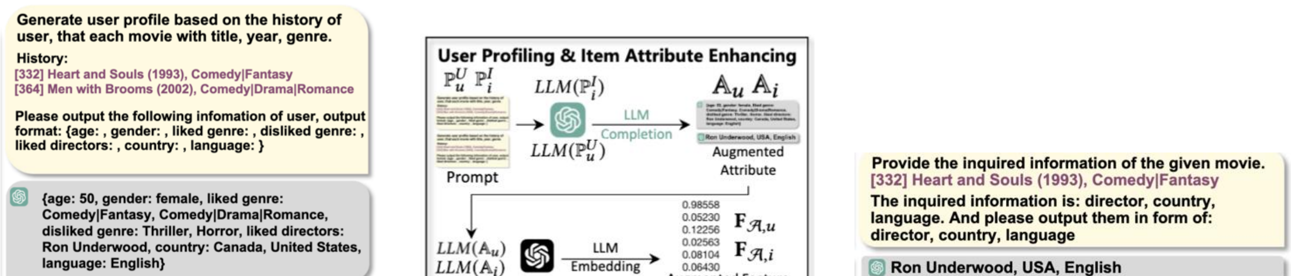


Leveraging LLMs to generate meaningful and effective embeddings for GNN's training

LLMs as Prefix

Embeddings from LLMs for GNNs

- LLMRec
  - Enrich the initial node embeddings for users and items with generated rich textual profiles
  - Achieve graph augmentation on user-item interaction data using GPT-3.5
  - Add meaningful training data
  - Ultimately improving the performance of recommenders



(b) User Profile (c) Item Attribute

Labels from LLMs for GNNs

- Leveraging the generated labels from LLMs as supervision
  - To improve the training of GNNs
- Generated information from LLMs is used
  - Not as input to the GNNs
  - Form the supervision signals for better optimization
- Enable GNNs to achieve higher performance on various tasks

- RLMRec model:

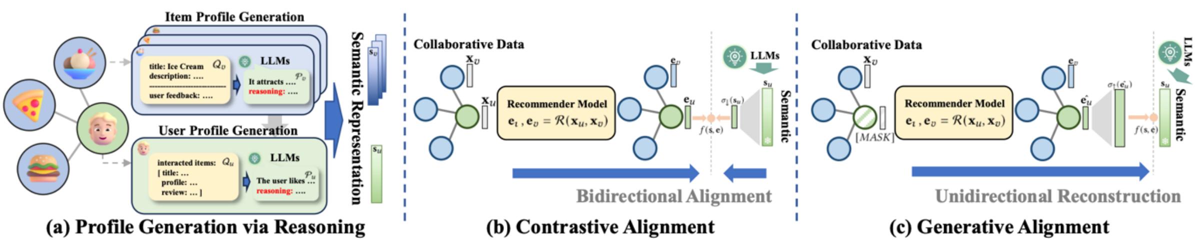


Figure 3: The overall framework of our proposed LLM-enhanced representation learning framework RLMRec.

Ren, Xubin, et al. "Representation learning with large language models for recommendation." *Proceedings of the ACM on Web Conference 2024*. 2024.

Results & Challenges

(LLMs as Prefix)

- **Generalization capability**
  - LLMs: generate meaningful and effective embeddings
- **Better optimization & higher performance**

- **Computational resource limitations**
- **Heavy dependency on LLMs**

LLMs-Graphs Integration

- Integrate LLMs with graph data
- Enhance (1) the ability of LLMs to tackle graph tasks and
  (2) the parameter learning of GNNs
- Categorized into three types:
  1. Fusion Training of GNNs and LLMs
  2. Alignment between GNNs and LLMs
  3. LLMs Agent for Graph

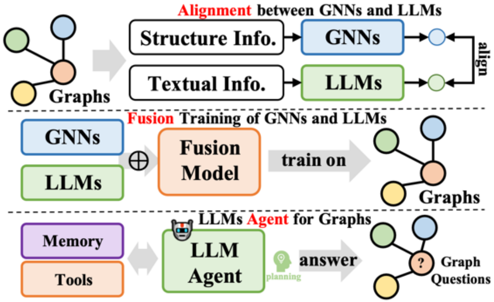


Figure 4: LLMs-Graphs Intergration.

Alignment between GNNs and LLM

- GNNs and LLMs are designed to handle different modalities of data
  - GNNs focusing on structural data and LLMs focusing on textual data
- This results in different feature spaces for the two modals
- To address this issue and make both modalities of data more beneficial for the learning of both GNNs and LLMs

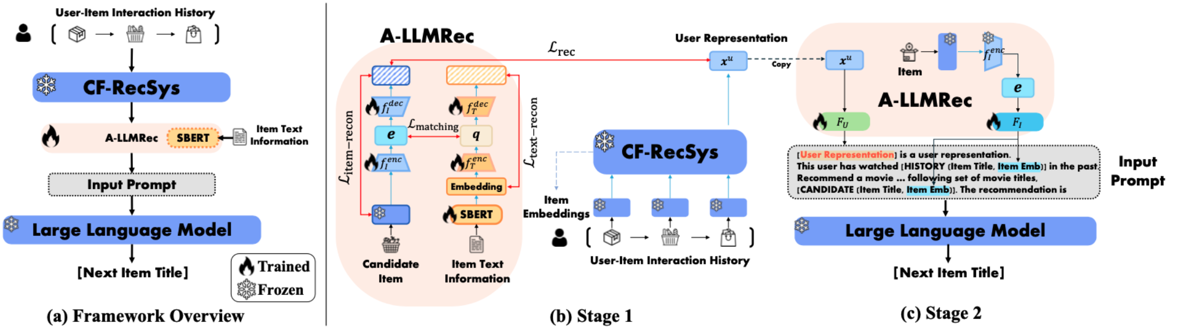
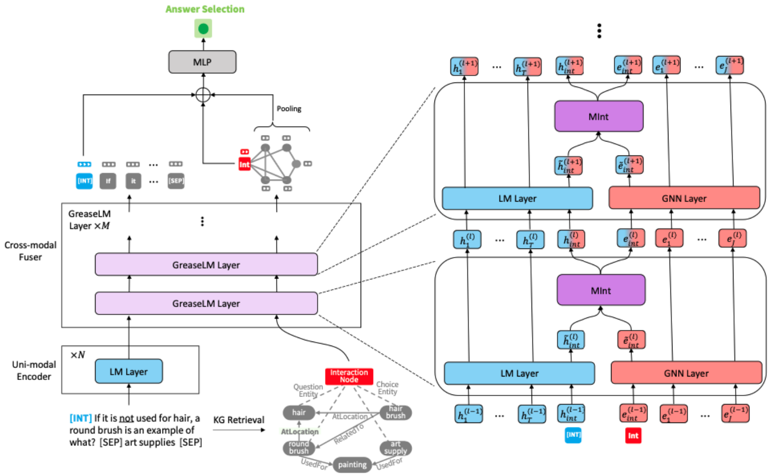


Figure 2: (a) is the overview of A-LLMRec. (b) and (c) are the detailed architecture of Stage 1 and Stage 2, respectively.

LLMs-Graphs Integration

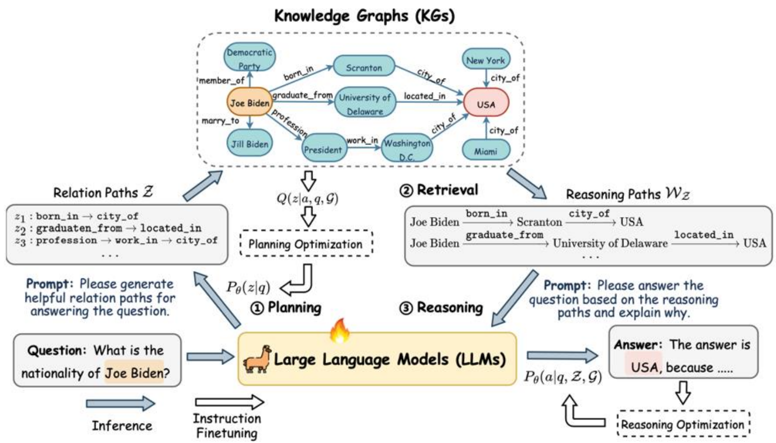
Fusion Training of GNNs and LLMs

- To achieve higher level of integration between LLMs and GNNs
- Designing a deeper fusion of the architecture of the modules
- Co-training GNNs and LLMs can result in win-win bi-directional benefit
- GreaseLM model:



LLMs Agent for Graph

- LLMs – powerful capabilities in understanding instructions and self-planning to solve tasks
- In graph domains, LLMs-based agents can interact directly with graph data
  - To perform tasks such as node classification and link prediction



Results
&
Challenges

(LLMs-Graphs Integration)

- Minimizing modality gap
- More accurate and flexible reasoning over graph data

- Scalability
- Limited interaction between graph agent and graph data

LLMs-Only

- Allow LLMs to directly accept graph structure information
- Understand it
- Perform inference for various downstream tasks
- Divided into two broad categories:
  1. Tuning-free
  2. Tuning-required

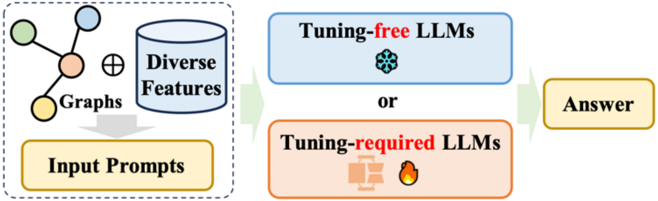


Figure 5: LLMs-Only.

Tuning-free

- Graph data has unique structured characteristics
- Two critical challenges arise:
  - effectively constructing a graph in natural language format
  - determining whether LLMs can accurately comprehend graph structures as represented linguistically
- To address these issues,
  - Tuning-free approaches are being developed to model and infer graphs solely within the text space
  - Exploring the potential of pre-trained LLMs for enhanced structural understanding

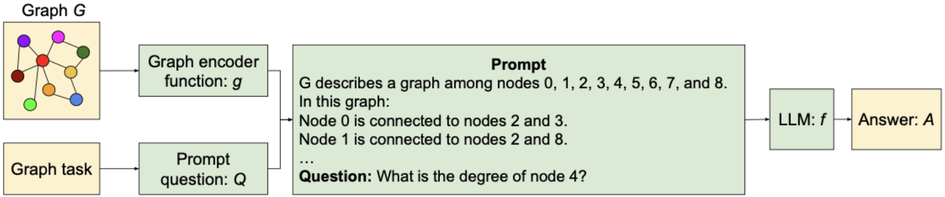
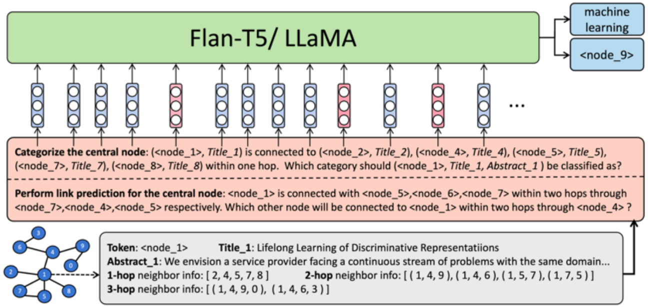


Figure 1: Overview of our framework for reasoning with graphs using LLMs.

LLMs-only

Tuning-required

- Limitations of expressing graph structural information using pure text
- To align graphs as node token sequences with natural language token sequences when inputting them to LLMs
- Tuning-required LLM-only approach
  - Discard the graph encoder and
  - Adopt a specific arrangement of graph token sequences
  - Along with carefully designed embeddings of graph tokens in prompts
  - Achieving promising performances in various downstream graph-related tasks



Results

&

Challenges

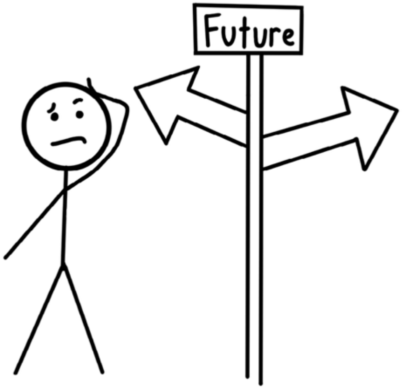
(LLMs-Only)

- Interpreting graph data
- Merging graphs with natural language instructions

- Effective transforming graphs into text prompts and reordering graph token sequences

Future Directions

- LLMs for Multi-modal Graphs
- Efficiency and Less Computational Cost
- Tackling Different Graph Tasks
- User-Centric Agents on Graphs

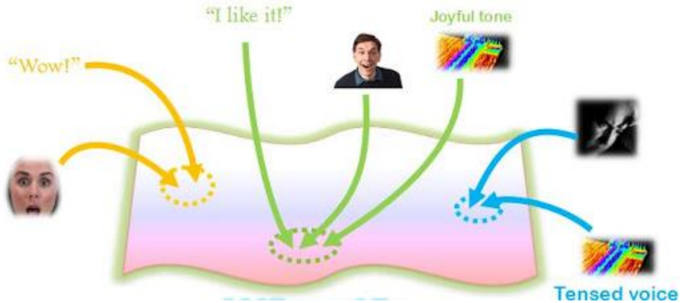


Future Directions

LLMs for Multi-modal Graphs

- To make LLMs process and understand multi-modal data
  - Nodes may contain features from multiple modalities
- By developing multi-modal LLMs
  - Enable more accurate and comprehensive reasoning over graph structures
  - Taking account not only textual information

    but also visual, auditory, and other types of data



Efficiency and Less Computational Cost

- The substantial computational expenses in LLMs pose a significant limitation
  - Associated with both the training and inference phases of LLMs

- Necessity to discover and implement efficient strategies
  - With reduced computational costs

Tackling Different Graph Tasks

- Potential in tackling more complex and generative tasks
  - Graph generation
  - Graph understanding
  - Graph-based question answering

- Able to unlock a large number of new opportunities for their applications across diverse domains
  - **Drug Discovery**: Generation of novel molecular structure
  - **Social Network Analysis**: Deeper insights into intricate relationship patterns
  - **Knowledge Graph Construction**: Creation of more comprehensive and contextually accurate knowledge bases

User-Centric Agents on Graphs

- LLM-based Agents:
  - Predominantly tailored for single graph tasks
  - One-time-run procedure

  - Should be user-friendly
  - Posses the capability to dynamically search for answers within graph data
    - In response to a diverse range of open-ended questions posed by users
  - Should be both adaptable and robust

Conclusion

- Integrating LLMs with graph learning techniques is a way to enhance performance in graph learning tasks

- Four main types of model architecture design for LLMs for graphs
  - GNNs as Prefix
  - LLMs as Prefix
  - LLMs-Graphs Integration
  - LLMs-only

- Future Directions
  - LLMs for Multi-modal Graphs
  - Efficiency and Less Computational Cost
  - Tackling Different Graph Tasks
  - User-Centric Agents on Graphs

Thank You!

HTET ARKAR
hak3601@cau.ac.kr