



LLMs for Graph

HTET ARKAR

Junior | Undergraduate

School of Computer Science and Engineering

Chung-Ang University

16-Jan-25

A Survey of Large Language Models for Graphs

Xubin Ren¹, Jiabin Tang¹, Dawei Yin², Nitesh Chawla³, Chao Huang¹

¹University of Hong Kong, ²Baidu Inc., ³University of Notre Dame

KDD '24

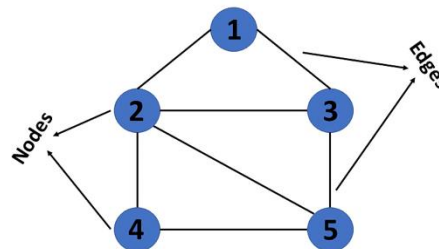
Contents

- ❖ Introduction
- ❖ Definitions
- ❖ LLMs for Graphs
 - *GNNs as Prefix*
 - *LLMs as Prefix*
 - *LLMs-Graphs Integration*
 - *LLMs-only*
- ❖ Future Directions
- ❖ Conclusion

Introduction

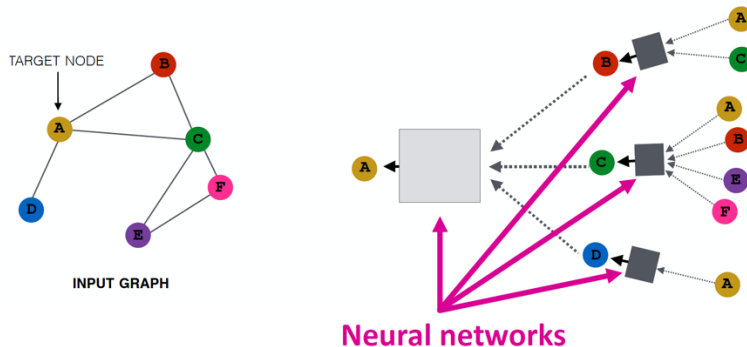
❖ Graphs

- **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



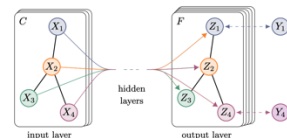
❖ Graph Neural Network (GNNs)

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



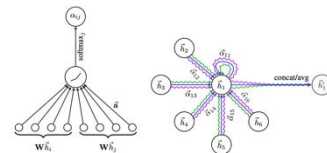
❖ Graph Convolutional Networks (GCNs)

- Effective in propagation embedding across nodes



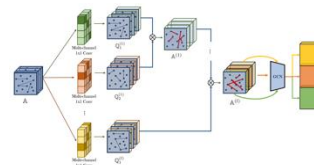
❖ Graph Attention Networks (GATs)

- Leverage attention mechanisms
- Perform precise aggregation of node features



❖ Graph Transformers

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



❖ Challenges

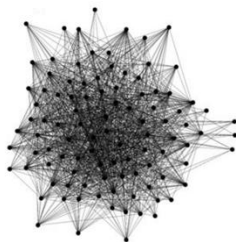
■ Data Sparsity

- Particularly in scenarios where graph structure is incomplete or noisy

■ Generalization ability

- New graphs or unseen nodes

❖ Need for more robust and adaptive models



Dense Graph

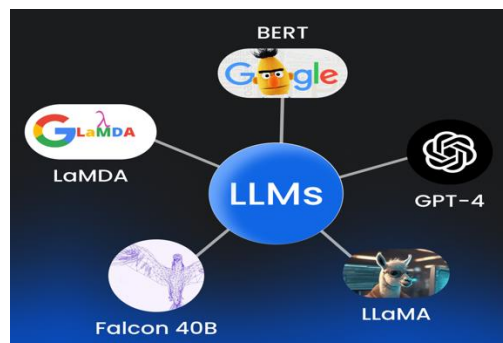


Sparse Graph

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



LLMs + GNNs \rightarrow 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

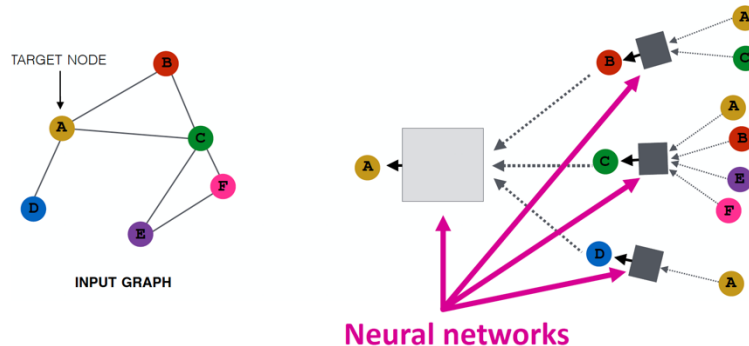
❖ Text-Attributed Graph (TAG)

- 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

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

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



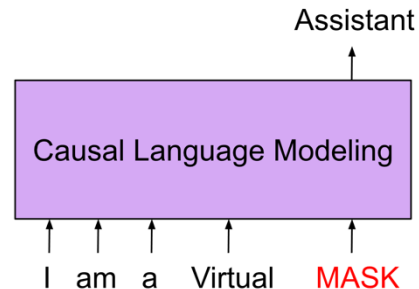
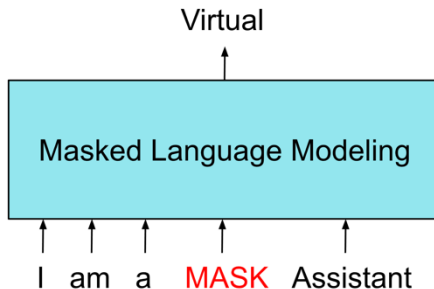
Large Language Models (LLMs)

- 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



Large Language Models (LLMs)

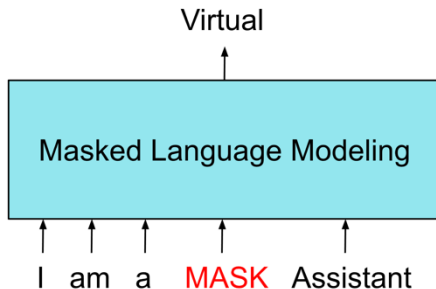
- Based on the direction of attention, LLMs can be categorized into two
 - (i) **Masked Language Modeling (MLM)**
 - (ii) **Causal Language Modeling (CLM)**



Masked Language Modeling (MLM)

- Making out certain tokens in a sequence
- Predicting the masked tokens **based on the surrounding context**
- Both left and right context of the masked token
- e.g., BERT, RoBERTa

"The quick brown [MASK] jumps over the lazy dog." → fox

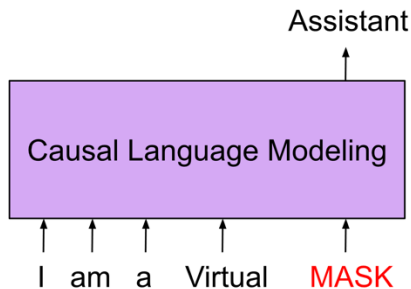


$$p(x_i | x_0, x_1, \dots, x_n)$$

Causal Language Modeling (CLM)

- Predicting the next token in a sequence **based on the previous tokens**
- Only the left context of the current token
- e.g., GPT and Llama

"The quick brown fox jumps" → **over**



$$p(x_i | x_0, x_1, \dots, x_{i-1})$$

LLMs for Graphs

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



LLMs for Graphs (Cont'd)

❖ GNNs as Prefix

- GNNs serve as the first component to process graph data
- Provide structure-aware tokens for LLMs for inference

❖ LLMs as Prefix

❖ LLMs-Graphs Integration

❖ LLMs-only

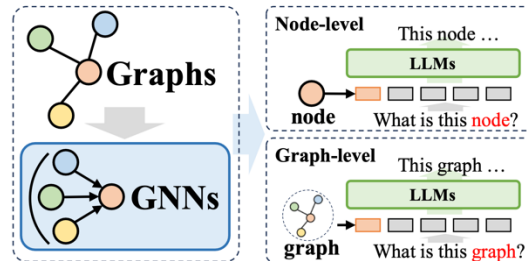


Figure 2: GNNs as Prefix.

LLMs for Graphs (Cont'd)

❖ GNNs as Prefix

❖ LLMs as Prefix

- LLMs first process graph data with textual information
- Provide node embeddings or generated labels for improved training of GNNs

❖ LLMs-Graphs Integration

❖ LLMs-only

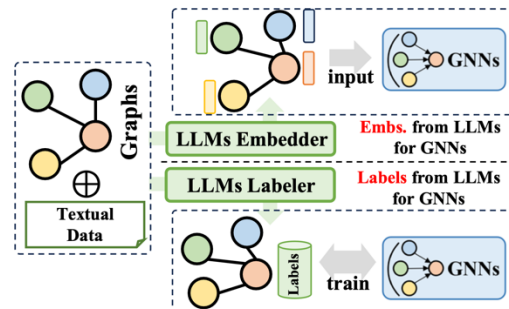


Figure 3: LLMs as Prefix.

LLMs for Graphs (Cont'd)

❖ GNNs as Prefix

❖ LLMs as Prefix

❖ LLMs-Graphs Integration

- LLMs achieve higher level of integration with graph data
- Fusion training
- Alignment with GNNs
- Building LLM-based agent
 - To interact with graph information

❖ LLMs-only

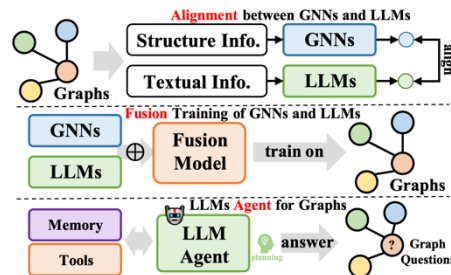


Figure 4: LLMs-Graphs Intergration.

LLMs for Graphs (Cont'd)

- ❖ GNNs as Prefix
- ❖ LLMs as Prefix
- ❖ LLMs-Graphs Integration
- ❖ LLMs-only
 - Designing practical prompting methods
 - Ground graph-structured data into sequences of words
 - Incorporate multi-model tokens



Figure 5: 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

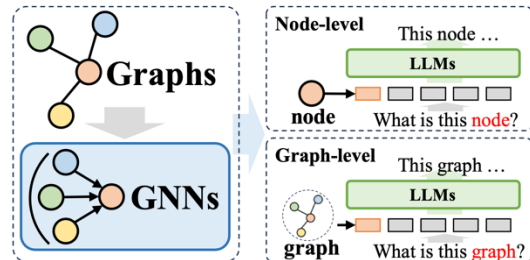
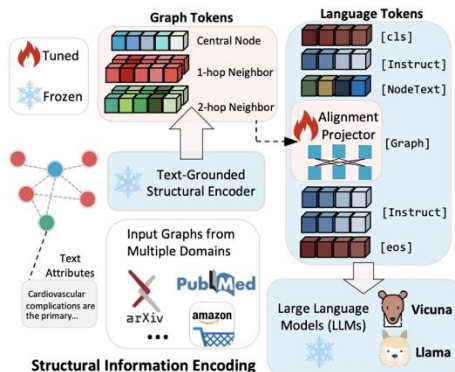


Figure 2: GNNs as Prefix.

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

<p>Graph Information: <graph>: Central Node: 68442, Edge index: [[...src node...], [...dst node...]], Node list: [...]</p> <p>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.</p> <p>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 ...</p>	Graph Matching
<p>Graph Information: <graph>: Central Node: 2, Edge index: [[...src node...], [...dst node...]], Node list: [...]</p> <p>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? ...</p> <p>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...</p>	Node Classification
<p>Graph Information: <graph>: Central Node 1: 8471, Edge index 1: [[...src node...], [...dst node...]], Node list 1: [...]</p> <p>Human Question: Given a sequence of graph tokens: <graph> that constitute a subgraph of a citation graph, ... Abstract: ... Title: ... and the other sequence of graph tokens: <graph>, ... Abstract: ... Title: ..., are these two central nodes connected? Give me an answer of "yes" or "no".</p> <p>GraphGPT Response: Yes, they are connected. Based on the first paper, ... And the second paper proposes ...</p>	Link Prediction

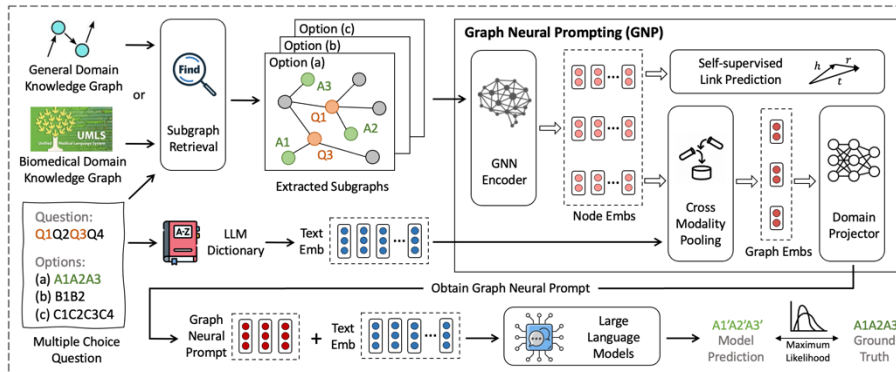
Figure 4: Our instruction designs for graph matching task (upper), node classification (middle) and link prediction (lower).

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



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

Question:

What is the best way to guess a babies eye color?

Options:

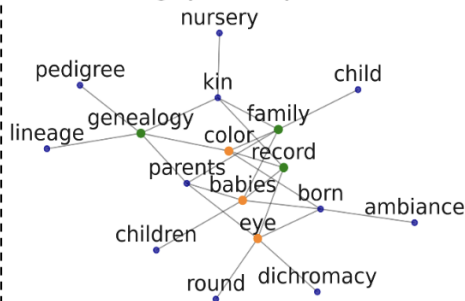
- (a) The surroundings they are born in
- (b) Their parents usual diet
- (c) Just take a random guess
- (d) The genealogy records of their family

Model Prediction:

Prompt Tuning: (c) Just take ... ❌

GNP: (d) The genealogy ... ✅

Retrieved subgraph from question entities:



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

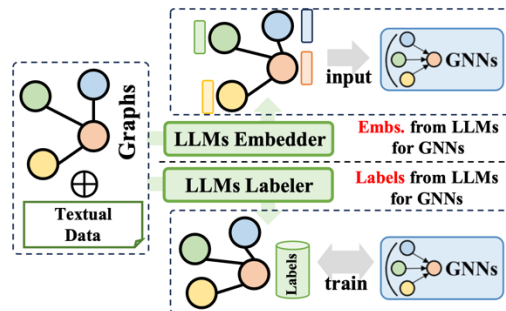
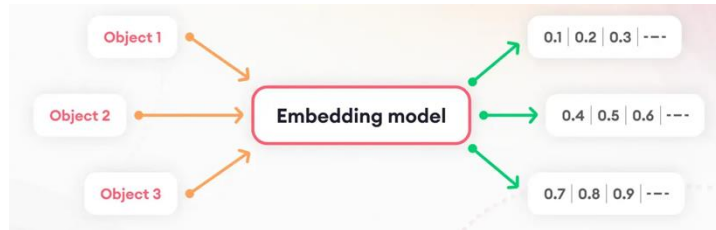


Figure 3: LLMs as Prefix.

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

Embeddings from LLMs for GNNs

➤ LLMRec

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

Generate user profile based on the history of user, that each movie with title, year, genre.

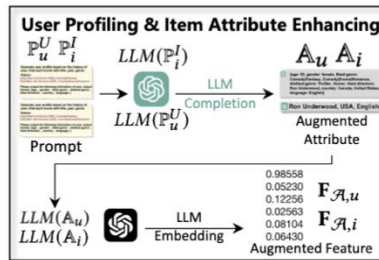
History:

[332] Heart and Souls (1993), Comedy|Fantasy
[364] Men with Brooms (2002), Comedy|Drama|Romance

Please output the following information of user, output format: {age: , gender: , liked genre: , disliked genre: , liked directors: , country: , language: }

{age: 50, gender: female, liked genre: Comedy|Fantasy, Comedy|Drama|Romance, disliked genre: Thriller, Horror, liked directors: Ron Underwood, country: Canada, United States, language: English}

(b) User Profile



Provide the inquired information of the given movie.
[332] Heart and Souls (1993), Comedy|Fantasy

The inquired information is: director, country, language. And please output them in form of: director, country, language

Ron Underwood, USA, English

(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, or
 - Form the supervision signals for better optimization
- Enable GNNs to achieve higher performance on various tasks

➤ RLMRec model:

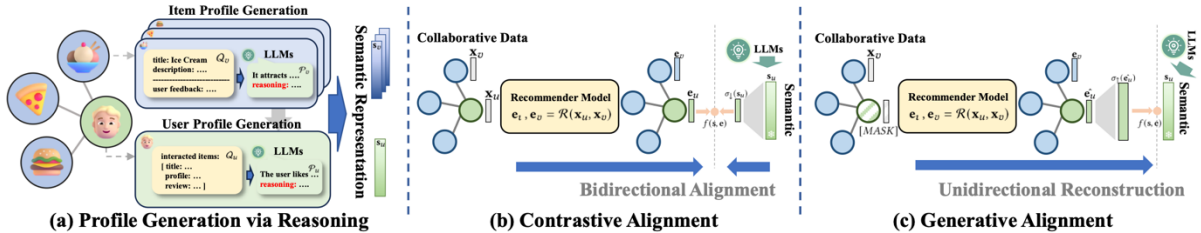


Figure 3: The overall framework of our proposed LLM-enhanced representation learning framework RLMRec.

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

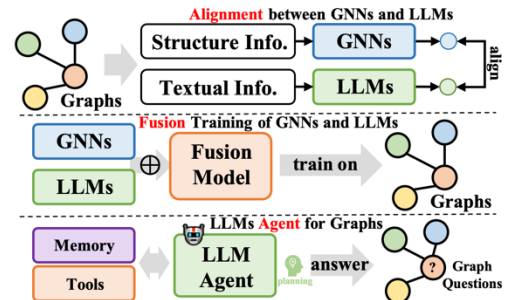
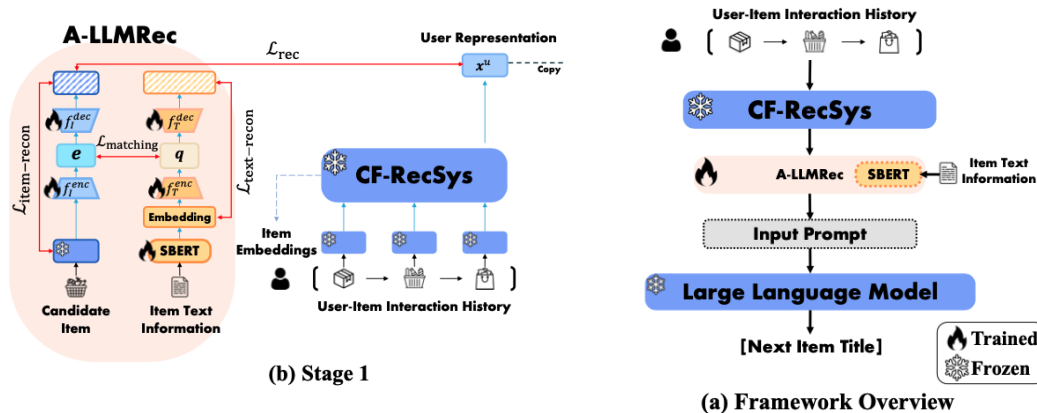


Figure 4: LLMs-Graphs Intergration.

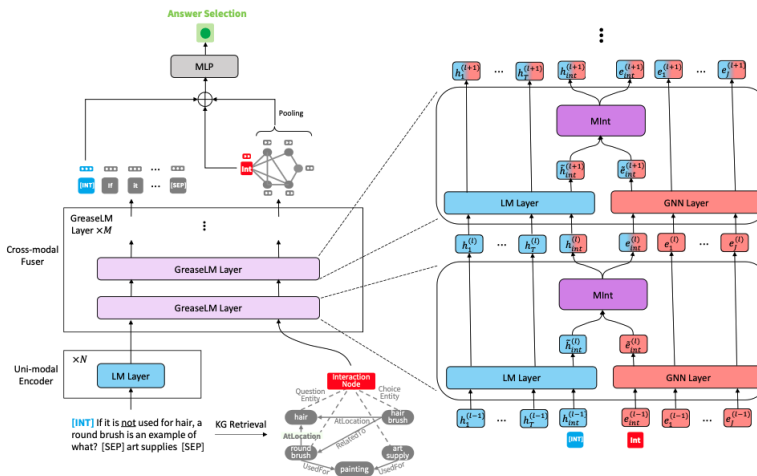
Alignment between GNNs and LLM

- 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 models
- To address this issue and make both modalities of data more beneficial for the learning of both GNNs and LLMs



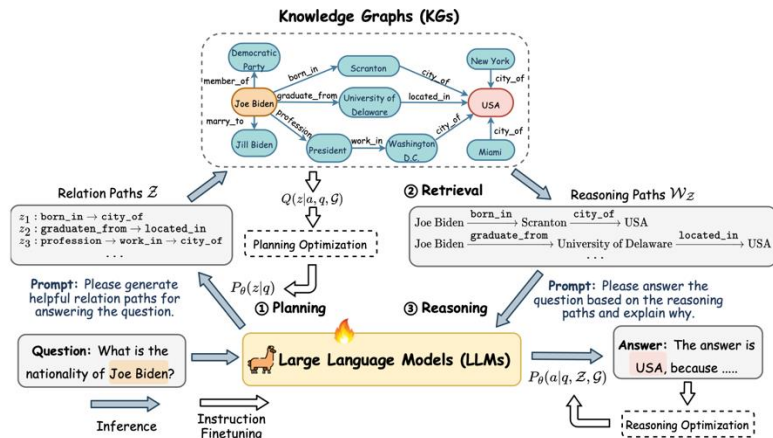
Fusion Training of GNNs and LLMs

- 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

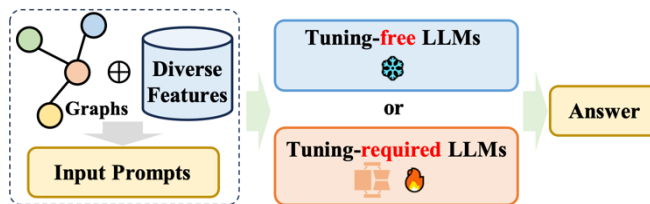


Figure 5: LLMs-Only.

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

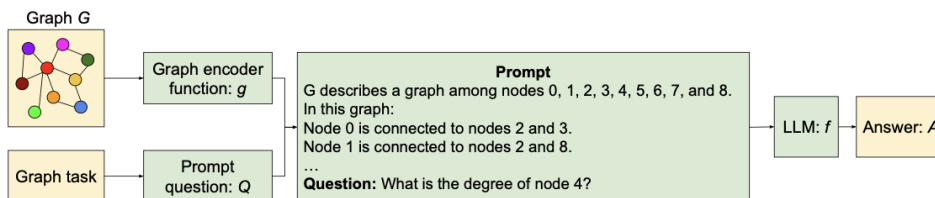
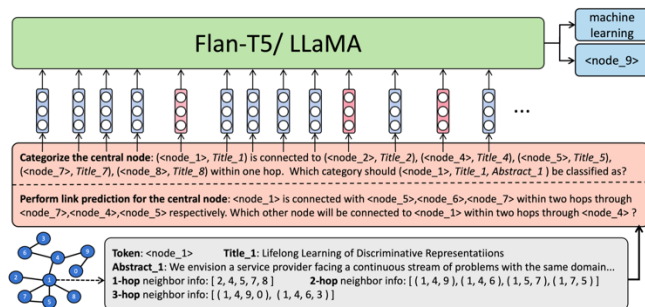


Figure 1: Overview of our framework for reasoning with graphs using LLMs.

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



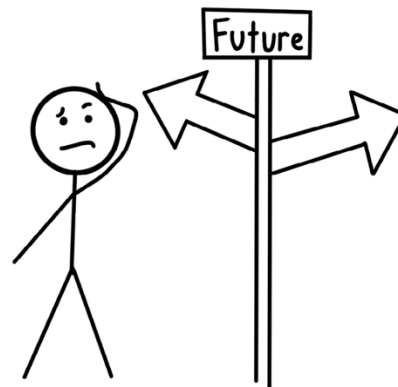
Results & Challenges

(LLMs-Only)

- ❖ Interpreting graph data
- ❖ Merging graphs with natural language instructions
- ❖ Effective transforming graphs into text prompts and reordering graph token sequences

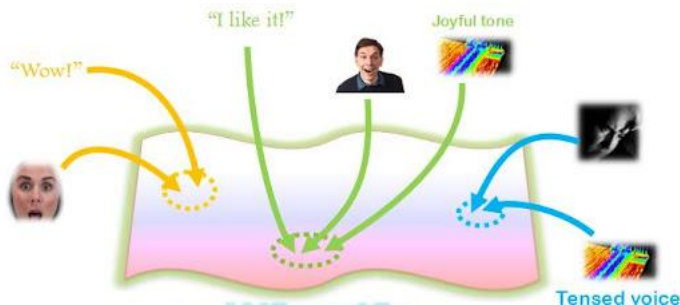
Future Directions

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



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

- **Graphs**: an essential data structure utilized to represent relationships in real-world scenarios
- Challenges like **data sparsity** and **limited generalization** capabilities continue to persist
- LLMs excel in language comprehension and summarization
- Integrating LLMs with graph learning techniques is a way to enhance performance in graph learning tasks

Conclusion

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

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

Thank You!



HTET ARKAR
hak3601@cau.ac.kr