Transforming Language Models: A Graph-based Approach to KR

Exploring the Fusion of Graph Structures and Language Models (LLMs)

Thandokuhle Brian Msane
Cape-KR 2024

Knowledge Representation Nature of Graphs

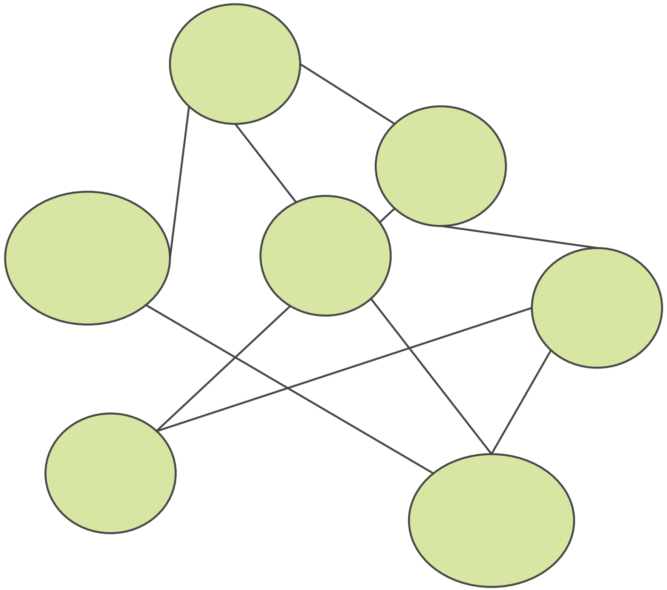
The advent of embedding graphs into LLMs came because of two main reasons:

1. **Guide the training** of LLMs based on the information represented on the graph.
2. **Serve as knowledge sources** to **enhance LLMs** and particularly alleviate **hallucinations**. Graphs are rich knowledge sources for the LLM.

Graphs in LLMs

The graphs that are used can be classified as follows:

a. **Pure graphs**
b. **Text-rich graphs**(node-/edge-level)
c. **Text-paired graphs**



LLMs on Graph Techniques (Text-rich)

Depending on the role of the LLM, we can have three methods that can be applied on such graphs:

1. **LLM as Predictor**

   These methods serve the ***language model as the main model architecture*** to capture both text information and graph structure information.

   Depending on how the graph structure information is injected into the models, they are classified into three types: **Graph as Sequence**, **Graph-empowered LLMs**, and **Graph-aware LLM fine-tuning** methods**.**

2. **LLM as Encoder**

   LLMs extract textual features to serve as initial node feature vectors for GNN, which generate node/edge representations and make predictions.

   Adopt **LLM-GNN cascaded architecture** to obtain the final representation which has textual information and structure information.

LLMs on Graph Techniques Cont'd

3. **LLM as Aligner**

The LLM and GNN components are **iteratively** trained together to enhance each other and depending on how they interact , during the training process, they can be classified into two types which are the;

- LLM-GNN **Prediction** Alignment,
- LLM-GNN **Latent Space** Alignment.

Graph Neural Networks (GNNs)

- A Deep Learning architecture for graph data
- Primarily designed to solve ***node-level tasks*** through the adoption of the **propagation-aggregation paradigm** to obtain node representation.

- Recently, GNNs to solve ***graph-level tasks*** have emerged.
- They adopt the **READOUT function** on node representations.
- These recent works can solve, issues such as ***over-smoothing***, ***over-squashing***, ***bias*** and ***interpretability***.

Graph Transformers (GTs)

- Techniques like **attention mechanisms**, **positional encoding, etc.,** are used within the layers of the transformers.
- They use the transformer as the base model architecture but inherently different from those transformers used in language models in three dimensions which are:
  - The tokens are **node tokens** yet on language models they are **word tokens.**
  - Their goal is to **encode the nodes or the entire graph** but language model transformers are meant **to encode text.**
  - Positional encoding is based on the **position of the word** for language model transformers but **graph transformers adopt shortest path distance, random walk distance to consider the distance of nodes in the graph.**

References

- ✓ Zhou J, et al. Graph neural networks: a review of methods and applications. AI Open. 2020;1(January):57–81. https://doi.org/10.1016/j.aiopen.2021.01.001.
- ✓ Jin, Bowen, et al. "Large Language Models on Graphs: A Comprehensive Survey." *arXiv preprint arXiv:2312.02783* (2023).