Assignment - 4

Name: Srilalitha Lakshmi Anusha Chebolu
Date: 02/20/25

New Insights:

GraphRag

1. uses LLM-generated knowledge graphs to provide improvements in QnA performance when conducting document analysis of complex information.
2. Limitations of Baseline Rag:
    1. struggles to connect the dots when anwering the question requires traversing disparate pieces on info through their shared attributes in order to provide new synthesized insights.
    2. performs poorly when being asked to holistically understand summarized semantic concepts over large collections of data.
    3. struggles with queries that require aggregation of information across the dataset to compose an answer.
    4. Queries such as "What are the top 5 themes in the data?" perform terribly because baseline RAG relies on a vector search of semantically similar text content within the dataset. There is nothing in the query to direct it to the correct information.
3. uses LLMs to generate KGs and this graph is then used alongside graph machine learning to perform prompt augmentation at query time.
    1. Graph ML : https://huggingface.co/blog/intro-graphml

In the example:

GraphRag approach discovered an entity in the query → Novorossiya

The LLM now ground to the entity in the graph and results in a superior answer that contains root through links to the original supporting text.

By using the LLM-generated knowledge graph, GraphRAG vastly improves the "retrieval" portion of RAG, populating the context window with higher relevance content, resulting in better answers and capturing evidence provenance.

Creating LLM-generated knowledge graphs:

Step 1:

LLM process → entire dataset,

creates references to → all entities and relationships

then used to create → KG by LLM

Step 2:

This KG graph is used to create → bottom-up clustering

The bottom -up clustering organizes → into semantic clusters

this clustering allows → pre summarization of semantic concepts and themes

Step 3:

At query time, both of these structures are used to provide materials for the LLM context window when answering a question.

what is bottom - up clustering : **Agglomerative clustering** is a bottom-up approach. It starts clustering by treating the individual data points as a single cluster, then it is merged continuously based on similarity until it forms one big cluster containing all objects.

Query

At query time, these structures are used to provide materials for the LLM context window when answering a question. The primary query modes are:

- *Global Search* for reasoning about holistic questions about the corpus by leveraging the community summaries.
- *Local Search* for reasoning about specific entities by fanning-out to their neighbors and associated concepts.
- *DRIFT Search* for reasoning about specific entities by fanning-out to their neighbors and associated concepts, but with the added context of community information.

Paper:

The approach uses an LLM to build a graph-based text index in two stages: first to derive an entity knowledge graph from the source documents, then to pre generate community summaries for all groups of closely-related entities.

Graph RAG Approach & Pipeline

2.1 Source Documents → Text Chunks

1. source documents should be split into text chunks for processing.
2. each of these chunks will be passed to a set of LLM prompts designed to extract the various elements of a graph index.

2.2 Text Chunks → Element Instances

1. identify and extract instances of graph nodes and edges from each chunk of source text.
2. a multipart LLM prompt
    1. first prompt:
        1. first identifies all entities in the text(name, type, description).

        2. then identifies all relationships between clearly-related entities (including source and target entities and a description of their relationship).
        3. both kinds of element instance are output in a single list of delimited tuples.
    2. second prompt is the covariate prompt that :
        1. extracts claims linked to detected entities which includes (subject, object, type, description, source text span, and start and end dates).
3. To ensure efficiency and quality → multiple rounds of "gleanings" → which encourages llms to detect any additional entities it may missed in prior extraction rounds.

2.3 Element Instances → Element Summaries

1. The use of LLM to "extract" descriptions of entities, relationships and claims in the source text is already in the form of abstractive summarization which is called instance level summary.
2. To convert all such instance level summaries into a single blocks of descriptive text for each graph element(entity node, edge and covariate) requires further round of LLM summarization over matching group of instances.

2.4 Element Summaries → Graph Communities

1. a variety of community detection algorithms may be used to partition the graph into communities of nodes with stronger connections to one another than to the other nodes in the graph.

used Leiden hierarchy to recover hierarchical community structure of large-scale graphs.

2.5 Graph Communities → Community Summaries

1. creates report-like summaries of each community in the Leiden hierarchy
2. These summaries are independently useful in their own right as a way to understand the global structure and semantics of the dataset, and may themselves be used to make sense of a corpus in the absence of a question.
3. Leaf-level communities. The element summaries of a leaf-level community (nodes, edges, covariates) are prioritized and then iteratively added to the LLM context window until the token limit is reached. The prioritization is as follows:
    1. for each community edge in decreasing order of combined source and target node degree (i.e., overall prominence), add descriptions of the source node, target node, linked covariates, and the edge itself.
4. Higher-level communities. If all element summaries fit within the token limit of the context window, proceed as for leaf-level communities and summarize all element summaries within the community. Otherwise, rank sub-communities in decreasing order of element summary tokens and iteratively substitute sub-community summaries (shorter) for their associated element summaries (longer) until fit within the context window is achieved.

2.6 Community Summaries → Community Answers → Global Answer

1. Prepare community summaries.
2. Map community answers.
3. Reduce to global answer.

REBEL:

Relation extraction by end to end language generation.

The aim of REBEL is to extract such triplets from raw text. This involves identifying entities and understanding the relationships between them, converting unstructured text into structured data that can be readily utilized for tasks like constructing knowledge bases.

Traditionally, NER + Relation Classification. Rebel attempt to combine these tasks into a single end-to end process.

Pre-trained → that allows for more efficient fine tuning.

significance of seq2seq model:

the model sees RE as a task where the input(text containing the entities) and the output(the generated relational triplet) are treated as sequences

can work with variable-length input sequences.

attention mechanisms helps the model capture long-distance dependencies between entities and their relationships in the text

The decoder takes the hidden states produced by the encoder and generates the output sequence, which represents the relational triplet(s). The decoder outputs the relation in an autoregressive fashion, meaning that it generates one part of the output at a time by conditioning on the previously generated parts. This helps in formulating coherent and contextually appropriate relationships based on the entities found in the input text.

Wikipedia as Knowledge Base:

- Contains extensive articles on diverse topics.
- Features hyperlinks connecting related entities.

Wikidata as Structured Database:

- Central repository for structured data about entities.
- Entities have unique identifiers and associated properties.

Matching Process:

- Hyperlinks in Wikipedia matched to corresponding Wikidata entities.
- Establishes a connection between unstructured text and structured data.

Extraction of Relational Information:

- Relationships identified when hyperlinks point to Wikidata entities.
- Triplets (subject, relation, object) extracted based on matched links.

Identifying Explicit Relationships:

- Enables extraction of clear, factual relationships mentioned in texts.
- Focuses on meaningful connections rather than ambiguous ones.

Creating the Dataset:

- Compiles extracted triplets to form a comprehensive dataset.
- Enables model training to recognize and extract relations from unstructured text.

Question:

1. Is there a tokenization method that preserves the full meaning of a token without breaking it into sub-tokens?

For example: I tried with SentencePiece tokenizer and it tokenized walked into ‘walk’ and ‘ed’ which looses its meaning rather it should be tokenized into ‘walked’. I guess this is same issue with BPE and wordPiece.