**(What)**

In our group project, what we are going to do is an interpretable personalized recommendation system based on Synergized LLM and KG reasoning.

**(Why KG + Recommendation)**

The traditional mainstream recommendation system (1995-2005) mainly includes three methods: content-based filtering method, collaborative filtering method and hybrid method. Although they solve the problem of information overload and long tail to some extent, traditional recommendation systems rely more on explicit or implicit feedback between users and items as input, which brings some drawbacks:

1. Data sparsity: In the actual scene, the interaction information between users and items is pretty sparse. For example, an online shopping app may contain hundreds of thousands of items, while the actual number of items purchased by the user may only be a couple hundreds. Using such a small amount of behavioral feedback data to predict a large amount of unknown information significantly increases the risk of overfitting and also affects the calculation of correlations in some algorithms, which makes it difficult to accurately predict user interests.

2. Cold start problem: For the recommendation of new users and new items, due to the lack of historical interaction information, the accuracy of the system recommendation will be greatly negatively affected.

3. Lack of diversity: Some traditional recommendation systems, such as content-based filtering recommendation methods, recommend items with high text relevance and can explain the recommendation results well, but the recommended results are often less surprising, or say lack of diversity.

4, Lack of generalization ability: the traditional collaborative filtering model lacks generalization ability, and is prone to head effect (popular items are over recommended) and tail effect (unpopular items are ignored).

A common approach to solving sparsity and cold start problems is to introduce additional auxiliary information into the input of the recommendation algorithm, which draws our attention to KG. On one hand, KG can provide rich domain knowledge as supplementary information to overcome the problems faced by collaborative filtering and content-based filtering recommendation methods (sparse and cold start problems); On the other hand, a recommendation system can use the semantic relationships present in KG to improve its accuracy, enable interpretability to results, and increase the variety of recommended items. Therefore, Incorporating knowledge graph into recommender systems has attracted increasing attention in recent years.

By introducing knowledge graphs, we can take advantage of knowledge graphs as follows to improve the accuracy, diversity, interpretability of recommendation systems:

1.Rich semantic information: The knowledge graph contains rich semantic information about the relationships between concepts, attributes, and entities. This helps to understand user interests more accurately. By exploring the interlinks within a knowledge graph, the connectivity between users and items can be discovered as paths, which provide rich and complementary information to user-item interactions. Such connectivity not only reveals the semantics of entities and relations, but also helps to comprehend a user’s interest. Specifically, KG recommendations leverage the connection between the entity representing the user, the item to be recommended, and their interaction. The recommendation system uses various connections to identify collections of items that may be of interest to the target user. As a result, complex relational representations provide additional valuable information for KG-based recommendation systems to apply reasoning between nodes to discover new connections. In contrast, generally the classical recommendation methods based on feature vectors ignore such connections, which may lead to poor overall recommendation performance, especially in the case of sparse data.

2.Interpretability: Interpretable recommendation systems are another hot research area in recent years. On one hand, if the implementation of the recommendation result presentation can provide appropriate recommendation explanation to the user, the user would better accept the recommendation result. On the other hand, we can also get a deeper understanding of the recommendation algorithm. KG just happens to contain a lot of auxiliary information that can be used to recommend the interpretation of results. For example, if the attention mechanism is applied to the embedding of relationships between entities in KG, then from the attention weight of different relationships, we can get the meaning of each type of item attribute to the target user. This technique can provide preference-level explanations for recommendations. Another example is that if we break down the relationship between the selected item and the target user or interactive item into a combination of several meta paths or meta graphs, interpretation can then be provided by transforming a metapath or metagram into understandable rules. What’s more, if the path embedding method is used, the weight of the specific path connecting the target user to the candidate item can be obtained through the attention mechanism. The weight of each path can represent the relative importance of each path to the user. Thus, explanations can be provided by generating explanations based on significant paths in the graph or interactive items in the multi-hop neighborhood.

3.Inference ability: Reasoning based on knowledge graph can be used for inconsistent detection, inference completion, knowledge discovery and other applications. For example, by training agents in UIKG using reinforcement learning techniques, it is possible to mine the actual paths that connect pairs of user items. It can directly show the reasoning process in KG, rather than looking for post-hoc explanations for recommended results. Therefore, the reasoning process is accurate and trustworthy for the target user.

**(Why LLM + KG)**

Then why we choose to combine LLM and KG in the recommendation system?

After summarizing their pros and cons, we found that their strengths and weaknesses can be very complementary. Therefore, unifying LLM and KGs together and leveraging their strengths at the same time is mutually reinforcing.

LLMs represent knowledge implicitly in their parameters, which means it is difficult to interpret or validate the knowledge obtained by LLMs. While KGs store explicit knowledge in a structural format (i.e., triples), which can be understandable by both humans and machines.

Most studies on KGs model the structure of knowledge, but ignore the textual information in KGs. The textual information in KGs is often ignored in KG-related tasks, such as KG completion and KGQA. LLMs have shown great performance in understanding natural language. Therefore, LLMs can be used in many textual information processing tasks, such as question answering, machine translation, and text generation.

LLMs are criticized for their lack of interpretability. It is unclear to know the specific patterns and functions LLMs use to arrive at predictions or decisions. But KGs are renowned for their symbolic reasoning ability, which provides an interpretable reasoning process that can be understood by humans.

Large language models (LLMs) have demonstrated impressive reasoning abilities in complex tasks. However, they lack up-to-date knowledge and experience hallucinations by generating content that while seemingly plausible but are factually incorrect. Knowledge graphs (KGs) offer a reliable source of knowledge for reasoning. Facts in KGs are usually manually curated or validated by experts, which are more accurate and dependable than those in LLMs.

KGs are hard to construct and often incomplete, which limits the ability of KGs to provide comprehensive knowledge. In comparison, LLMs enable great generalizability, which can be applied to various downstream tasks. By providing few-shot examples or finetuning on multi-task data, LLMs achieve great performance on many tasks.

LLMs trained on general corpus might not be able to generalize well to specific domains or new knowledge due to the lack of domain-specific knowledge or new training data. In contrast, many domains can construct their KGs by experts to provide precise and dependable domain-specific knowledge. Also the facts in KGs are continuously evolving. The KGs can be updated with new facts by inserting new triples and deleting outdated ones.

The pros and cons of KG and LLM are listed as follows for appendix and reference:

**KG的pros & cons：**

**Pros：**

Structural knowledge: KGs store facts in a structural format (i.e., triples), which can be understandable by both humans and machines.

Accuracy: Facts in KGs are usually manually curated or validated by experts, which are more accurate and dependable than those in LLMs.

Decisiveness果断: The factual knowledge in KGs is stored in a decisive manner. The reasoning algorithm in KGs is also deterministic, which can provide decisive results.

Interpretability: KGs are renowned for their symbolic reasoning ability, which provides an interpretable reasoning process that can be understood by humans.

Domain-specific knowledge: Many domains can construct their KGs by experts to provide precise and dependable domain-specific knowledge.

Evolving knowledge: The facts in KGs are continuously evolving. The KGs can be updated with new facts by inserting new triples and deleting outdated ones.

Explicit knowledge

Symbolic-reasoning

**Cons：**

Incompleteness: KGs are hard to construct and often incomplete, which limits the ability of KGs to provide comprehensive knowledge.

Lacking language understanding: Most studies on KGs model the structure of knowledge, but ignore the textual information in KGs. The textual information in KGs is often ignored in KG-related tasks, such as KG completion and KGQA.

Unseen facts: KGs are dynamically changing, which makes it difficult to model unseen entities and represent new facts.

Path uncertainty/non-uniqueness: Paths are uncertain between the kg subgraph of the training set and the point of the verification item. The description of kg would be one-sided or too long.

**LLM 的pros &cons：**

**pros:**

general knowledge: LLMs pre-trained on largescale corpora, which contain a large amount of general knowledge, such as commonsense knowledge and factual knowledge. Such knowledge can be distilled from LLMs and used for downstream tasks

language processing: LLMs have shown great performance in understanding natural language. Therefore, LLMs can be used in many natural language processing tasks, such as question answering, machine translation, and text generation.

generalizability: LLMs enable great generalizability, which can be applied to various downstream tasks. By providing few-shot examples or finetuning on multi-task data, LLMs achieve great performance on many tasks.

**cons：**

implicit knowledge: LLMs represent knowledge implicitly in their parameters. It is difficult to interpret or validate the knowledge obtained by LLMs.

Hallucination: LLMs often experience hallucinations by generating content that while seemingly plausible but are factually incorrect. This problem greatly reduces the trustworthiness of LLMs in real-world scenarios.

Indecisiveness: LLMs perform reasoning by generating from a probability model, which is an indecisive process. The generated results are sampled from the probability distribution, which is difficult to control.

black-box: LLMs are criticized for their lack of interpretability. It is unclear to know the specific patterns and functions LLMs use to arrive at predictions or decisions.

lacking domain-specific/new knowledge: LLMs trained on general corpus might not be able to generalize well to specific domains or new knowledge due to the lack of domain-specific knowledge or new training data.