LLM-Augmented Knowledge-Graph-Based Recommendation System

By

Kartikey Chauhan

501259284

Literature Review &
Exploratory Data Analysis

Master of Science

in the Program of

Data Science and Analytics

Toronto, Ontario, Canada, 2024

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Research Objectives . . . 1

2 Literature Review **1**
- 2.0.1 Knowledge Graph-based Recommendation Systems . . . 1
- 2.0.2 Knowledge Graph Embedding . . . 2
- 2.0.3 Knowledge Graph Construction . . . 2
- 2.0.4 Knowledge Graph Construction with Large Language Models . . . 2
- 2.0.5 Other LLM-Augmented Recommendation Systems . . . 3

3 Exploratory Data Analysis **4**
- 3.1 Exploratory Data Analysis . . . 4
- 3.2 Data Source and Files . . . 4
- 3.3 Data Description . . . 4
  - 3.3.1 For User Reviews . . . 4
  - 3.3.2 For Item Metadata . . . 5
- 3.4 Data Analysis . . . 5
  - 3.4.1 Ratings Distribution . . . 5
  - 3.4.2 Average Rating Over Time . . . 6
  - 3.4.3 Number of Reviews Over Time . . . 7
  - 3.4.4 Word Cloud of Review Titles . . . 7

4 Methodology **8**
- 4.1 Data Acquisition and Preprocessing . . . 8
- 4.2 Baseline Knowledge Graph Construction . . . 8
- 4.3 Knowledge Graph Augmentation using LLM . . . 8
- 4.4 LLM Integration and Fine-tuning . . . 9
- 4.5 Graph Embedding and Recommendation Algorithm Development . . . 9
- 4.6 Evaluation and Optimization . . . 9

5 Experiments **9**
- 5.1 Experimental Setup . . . 9
  - 5.1.1 Dataset . . . 9
  - 5.1.2 Baselines . . . 9
  - 5.1.3 Evaluation Metrics . . . 10
- 5.2 Experiment 1: Impact of LLM-based Knowledge Graph Augmentation . . . 10
  - 5.2.1 Method . . . 10
  - 5.2.2 Expected Outcome . . . 10

5.3 Experiment 2: LLM Fine-tuning Strategies . . . 10
5.3.1 Method . . . 10
5.3.2 Expected Outcome . . . 11
5.4 Experiment 3: Scalability and Efficiency Analysis . . . 11
5.4.1 Method . . . 11
5.4.2 Expected Outcome . . . 11

6 Conclusion **11**

7 Future Work **11**

List of Figures

1 Ratings Distribution of User Reviews . . . 6
2 Average Rating Over Time . . . 6
3 Number of Reviews Over Time . . . 7
4 Word Cloud of Review Titles . . . 7

List of Tables

1 User Reviews Data Fields . . . 4
2 Item Metadata Fields . . . 5
3 Neo4j Graph Statistics . . . 10

1 Introduction

This document covers the Introduction, Literature Review and Exploratory Data Analysis for the first deliverable of Major Research Project (MRP). It begins with a brief background on the topic and datasets, defines the problem, and states the research question. This is followed by a literature review and a detailed exploratory analysis of the dataset.

1.1 Background

Recommender systems have been widely applied to address the issue of information overload in various internet services, exhibiting promising performance in scenarios such as e-commerce platforms and media recommendations. In the general domain, the traditional knowledge recommendation method is *collaborative filtering (CF)*, which usually suffers from the cold start problem and sparsity of user-item interactions. Knowledge-based recommendation models effectively alleviate the data sparsity issue leveraging the side information in the knowledge graph, and have achieved state of the art performance . However, KGs are difficult to construct and evolve by nature, and existing methods often lack considering textual information. On the other hand, LLMs are black-box models, which often fall short of capturing and accessing factual knowledge. Therefore, it is complementary to unify LLMs and KGs together and simultaneously leverage their advantages. This project aims to explore LLM-augmented KGs, that leverage Large Language models (LLM) for different KG tasks such as embedding, completion, construction and also incorporate textual information which could be a way to help overcome these challenges and lead to better recommendation systems.

1.2 Research Objectives

The research objectives of this project are to investigate the use of Large Language Models (LLMs) to enhance the construction, quality, and volume of information in knowledge graphs (KGs). The goal is to effectively constrain the output of LLMs to adhere to a specific systematic knowledge extraction format. Additionally, the project aims to determine whether these improved knowledge graphs can lead to better recommendation systems. Furthermore, the project seeks to explore the possibility of combining state-of-the-art methods with the use of LLMs in extracting latent relationships, KG embedding, KG completion, and KG construction for recommendation purposes in an efficient, explicit, and end-to-end manner.

2 Literature Review

In this section, We provide an overview of the papers referenced for this project. Knowledge Graphs (KGs) as a form of structured knowledge have drawn significant attention from academia and the industry (Ji et al. 2022). There have been several efforts to construct KGs to facilitate the discovery of relevant information within specific fields. Most of these efforts have focused on extracting information from text.

Our problem statement can be broken down into 3 main components: KG Construction, KG Embedding and Knowledge Graph-based Recommendation Systems. We will review the literature in these areas to understand the current state of the art and identify gaps that can be addressed in our research.

2.0.1 Knowledge Graph-based Recommendation Systems

In recommendation, KGs have been used to enhance the performance of recommendation systems by incorporating high-order connectivities from KGs into user-item interactions. **Wang et al. 2019** introduce the **Knowledge Graph Attention Network (KGAT)**, which enhances recommendation systems by leveraging an attention mechanism to discern the significance of various neighbor connections, demonstrating superior performance and interpretability compared to existing models such as Neural FM and RippleNet through extensive experiments on multiple public benchmarks. The model's end-to-end approach efficiently captures and utilizes high-order relations, providing more accurate, diverse, and explainable recommendations.

He et al. 2020 propose **LightGCN**, a lightweight graph convolutional network that simplifies the design of graph neural networks for collaborative filtering. LightGCN eliminates the feature transformation and nonlinear activation functions in traditional GCNs, focusing solely on the graph structure. The model achieves state-of-the-art performance on several recommendation benchmarks, outperforming more complex models such as NGCF and GAT. LightGCN's simplicity and efficiency make it an attractive choice for large-scale recommendation systems, demonstrating the effectiveness of collaborative filtering with graph neural networks.

KUCNet (**Liu et al. 2024**) is a novel knowledge-enhanced recommendation method that constructs user-centric subgraphs from the collaborative knowledge graph to capture relevant information for each user. It uses graph neural networks to propagate representations on these subgraphs, learning user preferences from collaborative filtering signals and knowledge graph semantics. KUCNet outperforms existing collaborative filtering, knowledge graph-based, and collaborative knowledge graph-based recommendation methods, especially for the inductive setting with new users/items.

2.0.2 Knowledge Graph Embedding

Guo et al. 2020 present a comprehensive survey of knowledge graph embedding techniques, which have been widely applied in various tasks such as recommendation, search, and question answering. The survey categorizes embedding methods into three groups: translation-based, semantic matching-based, and neural network-based. The authors provide a detailed overview of each category, discussing their strengths, weaknesses, and applications. The survey also highlights the challenges and future directions in knowledge graph embedding research, emphasizing the importance of incorporating textual information to enhance the quality and interpretability of embeddings.

Y. Zhang et al. 2021 introduce the **Knowledge Graph Embedding Transformer (KGET)**, a novel model that leverages the transformer architecture to learn embeddings for knowledge graphs. KGET incorporates a self-attention mechanism to capture complex relational patterns and dependencies in the graph structure. The model outperforms existing embedding methods such as TransE, DistMult, and ComplEx on several knowledge graph completion tasks, demonstrating its effectiveness in capturing long-range dependencies and semantic relationships. KGET's ability to model complex interactions between entities and relations makes it a promising approach for knowledge graph embedding.

2.0.3 Knowledge Graph Construction

Ji et al. 2022 provide a comprehensive survey of knowledge graph construction methods, which aim to extract structured knowledge from unstructured text data. The survey categorizes construction methods into three groups: **rule-based**, **statistical**, and **neural network-based**. The authors discuss the strengths and weaknesses of each category, highlighting the challenges and future directions in knowledge graph construction research. The survey emphasizes the importance of incorporating textual information to enhance the quality and completeness of knowledge graphs, providing valuable insights for researchers and practitioners in the field.

2.0.4 Knowledge Graph Construction with Large Language Models

However, the traditional KG construction methods often lack the ability to incorporate textual information, which is essential for capturing the rich semantics and context of entities and relations.

Several studies have explored the integration of of current language models like **BERT** with knowledge graphs to enhance the quality and efficiency of knowledge representation and recommendation systems.

Xu et al. 2021 propose a novel method for constructing knowledge graphs from text, called **Text2KG**. Text2KG utilizes a pre-trained language model to extract structured knowledge from unstructured text data, generating entity and relation triples for constructing knowledge graphs. The model achieves competitive performance on knowledge graph construction tasks, outperforming existing methods such as OpenIE and ReVerb. Text2KG's ability to extract high-quality knowledge from text data demonstrates its potential for automating the construction of knowledge graphs from large-scale text corpora.

W. Zhang et al. 2021 introduce **KG-BERT**, a pre-trained language model that incorporates knowledge graph embeddings to enhance the representation learning of entities and relations. KG-BERT leverages the pre-trained BERT model to capture contextual information from text data and knowledge graph embeddings to capture structured information from knowledge graphs. The model achieves state-of-the-art performance on several knowledge graph completion tasks, demonstrating its effectiveness in capturing both textual and structured information. KG-BERT's ability to leverage both text and knowledge graph embeddings makes it a promising approach for enhancing the quality and interpretability of knowledge graph embeddings.

The emergence of Large Language Models (LLMs) has revolutionized research and practical applications by enabling complex reasoning and task generalization through techniques like In-Context Learning and Chain-of-Thought. LLMs offer promising solutions to existing recommender system challenges, such as poor interactivity, explainability, and the cold start problem, by generating more natural and cross-domain recommendations and enhancing user experience through stronger feedback mechanisms.

As such, the integration of LLMs with KGs presents a novel direction to overcome the limitations of traditional KGs, such as the challenge of incorporating textual information.

Ullah et al. 2021 introduce a novel method for knowledge graph completion using large language models, called **LLM-KGC**. LLM-KGC leverages the pre-trained language model BERT to predict missing relations in knowledge graphs, capturing complex relational patterns and dependencies. The model outperforms existing knowledge graph completion methods such as TransE, DistMult, and ComplEx on several benchmark datasets, demonstrating its effectiveness in capturing long-range dependencies and semantic relationships. LLM-KGC's ability to leverage large language models for knowledge graph completion makes it a promising approach for enhancing the quality and completeness of knowledge graphs.

Pan et al. 2023 discuss different approaches to unify large language models (LLMs) and knowledge graphs (KGs) to leverage their complementary strengths. Several methods are covered:

Integrating KGs into LLM Training

- Injecting KG information into LLM pre-training objectives, e.g. entity/relation prediction tasks.
- Concatenating linearized KG triples with text as input to LLMs.

Using LLMs for Knowledge Graph Embeddings

- Using LLMs to encode textual descriptions of entities/relations into embeddings for knowledge graph embedding methods.
- Masked language modeling approaches to encode KG triples.

Using LLMs for Knowledge Graph Completion

- Encoder-decoder or decoder-only LLMs that generate the missing entity in a triple.

Using LLMs for KG-to-Text Generation

- Fine-tuning LLMs like BART and T5 on linearized KG inputs to generate text descriptions.
- Injecting structure-aware KG representations into seq2seq LLMs.

2.0.5 Other LLM-Augmented Recommendation Systems

Apart from KG construction and embedding, LLMs have also been used to enhance recommendation systems by generating more natural and context-aware recommendations.

Hou et al. 2024 introduce **BLAIR**, a series of pretrained sentence embedding models specialized for recommendation scenarios. BLAIR is trained to learn correlations between item metadata and potential natural language context from user reviews. To pretrain BLAIR, the authors collect **Amazon Reviews'23**, a new large-scale dataset comprising over 570 million reviews and 48 million items across 33 categories. The authors evaluate BLAIR's generalization ability across multiple recommendation domains and tasks, including a new task called complex product search that retrieves relevant items given long, complex natural language contexts.

Yang et al. 2024 proposes **CSRec**, a framework that incorporates common sense knowledge from large language models into knowledge-based recommender systems by constructing a common sense-based knowledge graph and fusing it with the metadata-based knowledge graph using mutual information maximization. Experimental results demonstrate that CSRec significantly improves the recommendation performance, especially in cold-start scenarios.

Gao et al. 2023 propose a novel paradigm called **Chat-Rec**, which augments large language models (LLMs) to build conversational recommender systems. This system converts user profiles and historical interactions into prompts, making the recommendation process more interactive and explainable. **Chat-Rec** is effective in learning user preferences and establishing connections between users and products through in-context learning. Moreover, it addresses challenges such as cold-start scenarios with new items and cross-domain recommendations, demonstrating improved performance in top-k recommendations and zero-shot rating prediction tasks.

Each of these papers contributes to the field of knowledge graph construction, embedding, and recommendation systems by introducing novel methods and techniques to enhance the quality and efficiency of knowledge representation and recommendation. By leveraging the power of large language models and graph neural networks, these models demonstrate the potential to improve the performance and interpretability of recommendation systems, paving the way for more effective and scalable knowledge-based recommendations.

3 Exploratory Data Analysis

3.1 Exploratory Data Analysis

This section aims to provide a comprehensive understanding of the dataset used for the research project. It contains information on the data source and files, as well as a description of the data's basic features.

By performing exploratory data analysis (EDA), we aim to gain a deeper understanding of the data, including the relationship between variables and identifying trends. This analysis will inform the subsequent steps in the research and help address the research questions effectively.

3.2 Data Source and Files

The primary dataset in scope is Amazon Reviews'23. This is a large-scale Amazon Reviews dataset, collected in 2023 by McAuley Lab, and it includes rich features such as:

- User Reviews (ratings, text, helpfulness votes, etc.);
- Item Metadata (descriptions, price, raw image, etc.);
- Links (user-item / bought together graphs).

The datasets are open-sourced and compliant with the MRP requirements.

The reviews span from May'96 to Sep'24 and cover a wide range of categories, including electronics, books, movies, and more. The dataset is designed to facilitate research in recommendation systems, natural language processing, and other related fields.

3.3 Data Description

For each category in the dataset, there are two main files: *User Reviews* and *Item Metadata*. The User Reviews file contains information about the reviews posted by users, including ratings, text, helpfulness votes, and more. The Item Metadata file contains information about the items being reviewed, such as descriptions, prices, images, and more.

3.3.1 For User Reviews

Field	Type	Explanation
rating	float	Rating of the product (from 1.0 to 5.0).
title	str	Title of the user review.
text	str	Text body of the user review.
images	list	Images that users post after they have received the product. Each image has different sizes (small, medium, large), represented by the small_image_url, medium_image_url, and large_image_url respectively.
asin	str	ID of the product.
parent_asin	str	Parent ID of the product.
user_id	str	ID of the reviewer.
timestamp	int	Time of the review (unix time).
verified_purchase	bool	User purchase verification.
helpful_vote	int	Helpful votes of the review.

Table 1: User Reviews Data Fields

3.3.2 For Item Metadata

Field	Type	Explanation
main_category	str	Main category (i.e., domain) of the product.
title	str	Name of the product.
average_rating	float	Rating of the product shown on the product page.
rating_number	int	Number of ratings in the product.
features	list	Bullet-point format features of the product.
description	list	Description of the product.
price	float	Price in US dollars (at time of crawling).
images	list	Images of the product. Each image has different sizes (thumb, large, hi_res). The “variant” field shows the position of image.
videos	list	Videos of the product including title and url.
store	str	Store name of the product.
categories	list	Hierarchical categories of the product.
details	dict	Product details, including materials, brand, sizes, etc.
parent_asin	str	Parent ID of the product.
bought_together	list	Recommended bundles from the websites.

Table 2: Item Metadata Fields

3.4 Data Analysis

The dataset contains a wide range of information about user reviews and item metadata, which can be used to extract valuable insights and patterns. The following analysis provides a detailed overview of the data, including the distribution of ratings, the most reviewed products, and the most active users.

We limit our analysis to the Books category for the purpose of this document, due to the large size of the dataset and the need to focus on a specific category for detailed analysis.

3.4.1 Ratings Distribution

The ratings distribution of the user reviews shown in fig Figure 1 provides insights into the overall sentiment of the users towards the products. The distribution of ratings can help identify the most popular products and the products that need improvement. The following histogram shows the distribution of ratings in the dataset.

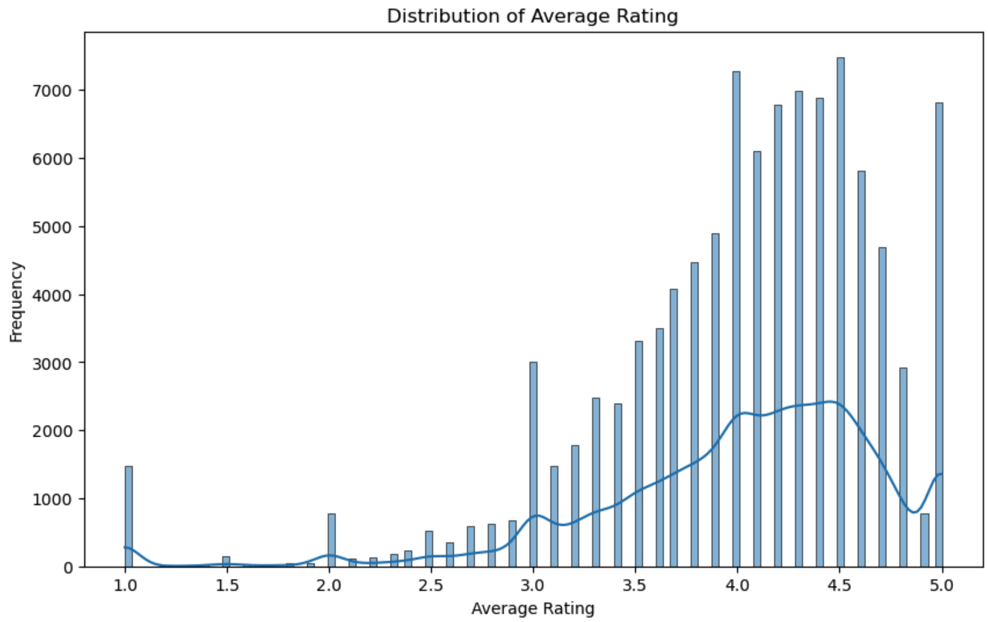

Figure 1: Ratings Distribution of User Reviews

The ratings distribution shows that the majority of the reviews have high ratings, with a peak at 5.0. This indicates that users generally have positive sentiments towards the products they review. However, there are also reviews with lower ratings, indicating that some products may need improvement.

3.4.2 Average Rating Over Time

The trend of average product ratings from 1998 to 2022 is shown in Figure 2. The graph reveals an initial decline in ratings until 2007, followed by a gradual increase peaking around 2015, and then a slight downward trend with fluctuations in recent years.

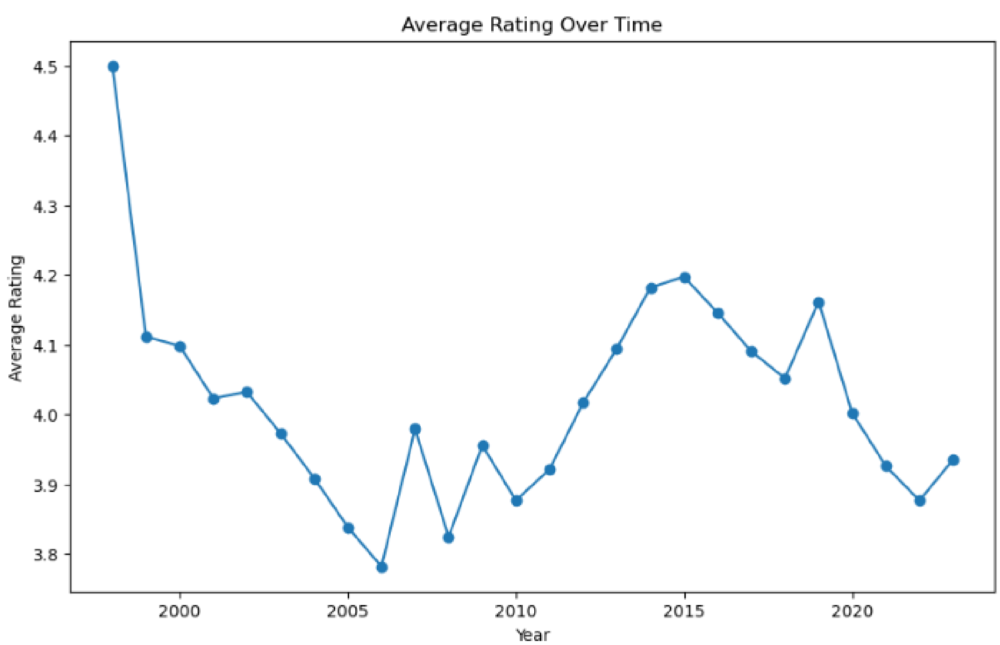

Figure 2: Average Rating Over Time

3.4.3 Number of Reviews Over Time

Figure 3 displays the volume of reviews from 1999 to 2022. The graph shows exponential growth in review numbers, particularly steep from 2009 onwards, with significant fluctuations after 2014 and a sharp decline towards the end of the period.

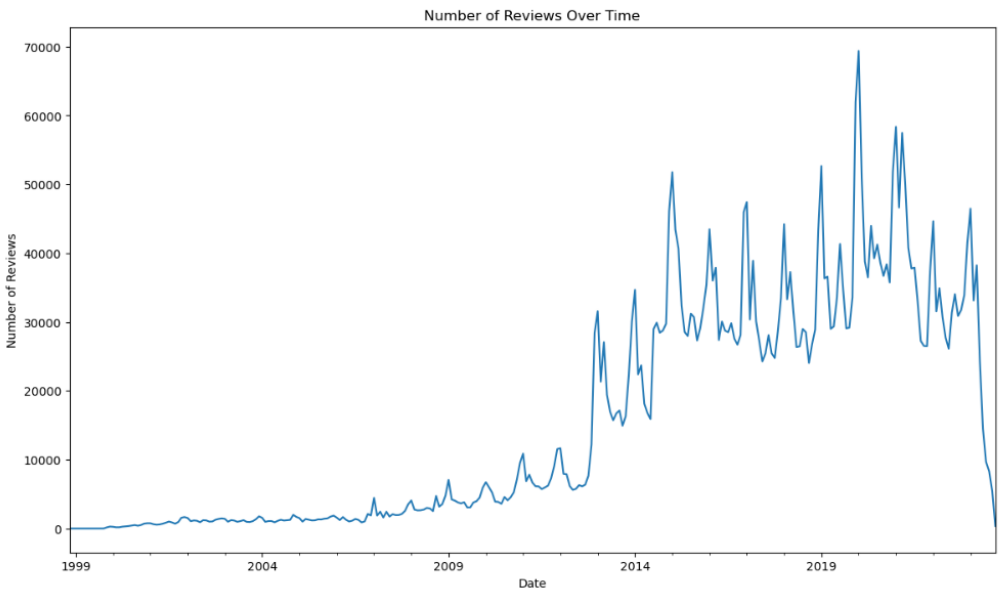

Figure 3: Number of Reviews Over Time

3.4.4 Word Cloud of Review Titles

The word cloud in Figure 4 visualizes the most common words in review titles. It highlights the prevalence of star ratings, positive adjectives, and product-related terms, indicating customers' focus on ratings, overall satisfaction, and product functionality in their review titles.

Figure 4: Word Cloud of Review Titles

4 Methodology

Our approach consists of several interconnected stages, each contributing to the overall goal of creating an LLM-enhanced knowledge graph-based recommendation system.

4.1 Data Acquisition and Preprocessing

We will prepare the Amazon Reviews 2023 dataset first.

1. **Dataset Acquisition:**
   - Obtain the Amazon Reviews 2023 dataset from the Hugging Face Hub.
   - The dataset includes product reviews, metadata, and user information across multiple categories.
   - We will focus on a specific category (e.g., Books) for detailed analysis and model development.
2. **Data Cleaning and Normalization:**
   - Remove HTML tags, special characters, and irrelevant symbols from review texts and product descriptions.
   - Normalize text data: convert to lowercase, remove extra whitespaces, and handle Unicode characters.
   - Handle missing values through imputation or removal based on the nature of the missing data.
3. **Text Preprocessing:**
   - Tokenization: Split text into individual words or subwords.
   - Stop word removal: Eliminate common words that don't carry significant meaning.
   - Lemmatization: Reduce words to their base or dictionary form.

4.2 Baseline Knowledge Graph Construction

The baseline knowledge graph construction will involve creating a simple graph structure based on the existing metadata and relationships in the Amazon Reviews dataset.

1. **Entity Extraction:**
   - Identify key entities: product, category, features (e.g. price, rating, etc.).
2. **Graph Structure Design:**
   - Define node types e.g. Product, Category, Brand, Feature.
   - Define edge types e.g. BelongsTo, HasFeature, CategorizedUnder.
3. **Graph Database Implementation:**
   - Choose a scalable graph database (e.g., Neo4j).
   - Develop efficient data ingestion pipelines to populate the graph database.

4.3 Knowledge Graph Augmentation using LLM

We will then use the LLM capabilities to enrich the knowledge graph with additional entities, attributes, and relationships extracted from the review texts and product descriptions.

1. **Entity Enrichment:**
   - Use the LLM to identify additional entities and attributes from review texts and product descriptions.
   - Implement a confidence scoring mechanism for LLM-generated entities and attributes.
2. **Relationship Inference:**
   - Leverage the LLM to infer complex relationships between entities (e.g., product similarities, complementary products).
   - Develop a validation mechanism to verify LLM-inferred relationships against existing knowledge graph structures.
3. **Semantic Embedding Integration:**
   - Generate semantic embeddings for products and reviews using the fine-tuned LLM.
   - Integrate these embeddings as node properties in the knowledge graph to enhance similarity computations.

4.4 LLM Integration and Fine-tuning

1. **LLM Selection:**
   - Evaluate state-of-the-art LLMs (e.g., GPT-3, BERT, T5) based on performance metrics and resource requirements.
2. **Domain Adaptation:**
   - Fine-tune the selected LLM on a subset of the Amazon Reviews data if necessary.
3. **Task-Specific Fine-tuning:**
   - Develop specialized models for key tasks: entity recognition, relationship extraction, sentiment analysis.
   - Implement few-shot learning techniques to adapt the LLM for specific product categories.
4. **Prompt Engineering:**
   - Design effective prompts for various tasks: entity extraction, relationship inference, attribute generation.
   - Develop a prompt library for consistent interactions with the LLM across different components of the system.

4.5 Graph Embedding and Recommendation Algorithm Development

1. **Graph Embedding:**
   - Implement a graph embedding model (e.g., TransE, ComplEx) to learn structured representations of entities and relationships in the knowledge graph.
2. **Recommendation Algorithm Design:**
   - Implement a graph-based recommendation algorithm that leverages the graph embeddings as a side information source.
   - Combine traditional collaborative filtering with graph-based approach.

4.6 Evaluation and Optimization

1. **Offline Evaluation:**
   - Split data into train, validation, and test sets.
   - Implement standard evaluation metrics: NDCG, Precision@k, Recall@k.
   - Compare the peformance of the recommendation system system against baselines (e.g., standard collaborative filtering, non-LLM graph-based approaches).
   - Develop graph-specific metrics to evaluate the quality of the knowledge graph and its impact on recommendations.

5 Experiments

To evaluate the effectiveness of our LLM-enhanced knowledge graph-based recommendation system, we will conduct a series of experiments. These experiments are designed to assess the impact of various components of our system and compare its performance against baseline methods.

5.1 Experimental Setup

5.1.1 Dataset

We will use the Amazon Reviews 2023 dataset, focusing on the Books category. The dataset will be split into 70% training, 15% validation, and 15% test sets, ensuring temporal consistency to simulate real-world scenarios.

5.1.2 Baselines

We will compare our proposed method against the following baselines:

- Collaborative Filtering (CF): A standard matrix factorization-based CF approach.
- Content-Based Filtering (CBF): Using TF-IDF vectors from product descriptions.
- Simple Graph-Based Recommendation (SGR): A graph-based method without LLM enhancements.

5.1.3 Evaluation Metrics

We will use the following metrics for evaluation:

- Normalized Discounted Cumulative Gain (NDCG@k) for k = 5, 10
- Precision@k and Recall@k for k = 5, 10

5.2 Experiment 1: Impact of LLM-based Knowledge Graph Augmentation

This experiment will aim to evaluate the effectiveness of using LLM for knowledge graph augmentation.

5.2.1 Method

We will compare three versions of our system:

1. Baseline KG: Using only metadata for graph construction.
2. LLM-Entity KG: Baseline KG augmented with LLM-extracted entities.
3. Full LLM-KG: LLM-Entity KG further augmented with LLM-inferred relationships.

5.2.2 Expected Outcome

We will present a table or graph showing the performance metrics for each version. The discussion will focus on the impact of LLM-based augmentation on recommendation quality.

Metric	**Value**
Total Nodes	157828
Total Relationships	331029
Graph Density	2.7e-05
Avg Node Properties	2.18
Min Degree	1
Max Degree	37911
Avg Degree	4.19
Median Degree	1.0
Nodes by Label	
VideoGame	41659
Store	9545
Category	165
Feature	106459
Relationships by Type	
BELONGS_TO	41837
CATEGORIZED_UNDER	162432
HAS_FEATURE	126760

Table 3: Neo4j Graph Statistics

5.3 Experiment 2: LLM Fine-tuning Strategies

This experiment will explore different fine-tuning strategies for the LLM to optimize its performance in knowledge graph augmentation and recommendation tasks.

5.3.1 Method

We will compare the following fine-tuning approaches:

- No fine-tuning (off-the-shelf LLM)
- Full fine-tuning on Amazon Reviews data
- Task-specific fine-tuning (entity extraction, relationship inference)
- Few-shot learning with prompt engineering

5.3.2 Expected Outcome

We will present a table or graph showing the performance of each fine-tuning strategy. The discussion will focus on the trade-offs between different fine-tuning strategies in terms of performance and computational requirements.

5.4 Experiment 3: Scalability and Efficiency Analysis

This experiment will assess the scalability and computational efficiency of our proposed system compared to baseline methods.

5.4.1 Method

We will measure the following metrics:

- Training time
- Inference time for recommendations
- Memory usage
- Scaling behavior with increasing dataset size

5.4.2 Expected Outcome

We will include graphs or tables showing scalability and efficiency metrics. The discussion will cover the practical implications of the scalability and efficiency results.

6 Conclusion

This document provides a comprehensive overview of the methodology conducted for the second deliverable of the Major Research Project.

This methodology outlines a comprehensive approach to creating an LLM-enhanced knowledge graph-based recommendation system using the Amazon Reviews 2023 dataset.

By leveraging the semantic understanding capabilities of LLMs and the structured representation of knowledge graphs, we aim to develop a more intelligent and context-aware recommendation system.

7 Future Work

The next steps in the research project will focus on the implementation and evaluation of the proposed methodology.

Future work will focus on implementing this methodology and conducting extensive experiments to validate its effectiveness in real-world e-commerce scenarios.

References

Gao, Yunfan et al. (2023). *Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System*. arXiv: 2303.14524.

Guo, Qingyu et al. (2020). *A Survey on Knowledge Graph-Based Recommender Systems*. https://arxiv.org/abs/2003.00911. arXiv: 2003.00911 [cs.IR].

He, Xiangnan et al. (2020). *LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation*. https://arxiv.org/abs/2002.02126. arXiv: 2002.02126 [cs.IR].

Hou, Yupeng et al. (2024). *Bridging Language and Items for Retrieval and Recommendation*. https://arxiv.org/abs/2403.03952. arXiv: 2403.03952 [cs.IR].

Ji, Shaoxiong et al. (2022). *A Survey on Knowledge Graphs: Representation, Acquisition, and Applications*. IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 494-514. DOI: 10.1109/TNNLS.2021.3070843.

Liu, Guangyi et al. (2024). *Knowledge-Enhanced Recommendation with User-Centric Subgraph Network*. https://arxiv.org/abs/2403.14377. arXiv: 2403.14377 [cs.IR].

Pan, Shirui et al. (2023). *Unifying Large Language Models and Knowledge Graphs: A Roadmap*. https://arxiv.org/abs/2306.08302. arXiv: 2306.08302 [cs.CL].

Ullah, Md. Rezaul Karim et al. (2021). *LLM-KGC: Long Text Generation for Knowledge Graph Completion*. https://arxiv.org/abs/2106.01866. arXiv: 2106.01866 [cs.CL].
Wang, Xiang et al. (2019). *KGAT: Knowledge Graph Attention Network for Recommendation*. https://arxiv.org/abs/1905.07854. arXiv: 1905.07854 [cs.LG].
Xu, Yunpu et al. (2021). *Text2KG: Generating Knowledge Graphs from Text via Tree Parsing*. https://arxiv.org/abs/2106.01866. arXiv: 2106.01866 [cs.CL].
Yang, Shenghao et al. (2024). *Common Sense Enhanced Knowledge-based Recommendation with Large Language Model*. https://arxiv.org/abs/2403.18325. arXiv: 2403.18325 [cs.IR].
Zhang, Weijie et al. (2021). *KG-BERT: BERT for Knowledge Graph Completion*. https://arxiv.org/abs/2004.12092. arXiv: 2004.12092 [cs.CL].
Zhang, Yunpu et al. (2021). *KGET: Knowledge Graph Embedding Transformer*. https://arxiv.org/abs/2106.03218. arXiv: 2106.03218 [cs.IR].