Large Language Models for Recommender Systems

jh241004

Representative: Toyotaro Suzumura (The University of Tokyo)

Members: Hiroki Kanezashi, Refaldi Putra, Igor Lima Rocha Azevedo, Junyi Chen, Limin Wang

Overview

Background: Recommender systems have been integrated into e-commerce platforms, news websites, and other online services.

- Personalized Recommendations: Tailored to reflect individual user behavior.
- Awareness of Social Issues: Addressing concerns such as privacy, filter bubbles, and fairness.

Many methods utilize neural network models like Graph Neural Networks (GNN) and Transformers to capture complex user behaviors and hidden intentions.

- Recently, Large Language Models (LLMs) and other Transformer-based models can leverage text datasets, such as descriptions and user reviews.
- However, the application of LLMs in recommender systems is still developing, with unresolved issues including improving recommendation accuracy and comparing their effectiveness to existing models like GNNs and Transformers.

Research Goal: Explore models and algorithms that can enhance LLM-based recommender systems.

1. Utilize LLMs to predict the sustainability of interest and relevance of items and make recommendations based on these predictions.
2. Optimize the encoding of numerical information for LLM-based recommender systems.
3. Develop a Multi-Behavior Recommendation model based on LLMs.

Unified Recommender Model for E-Commerce

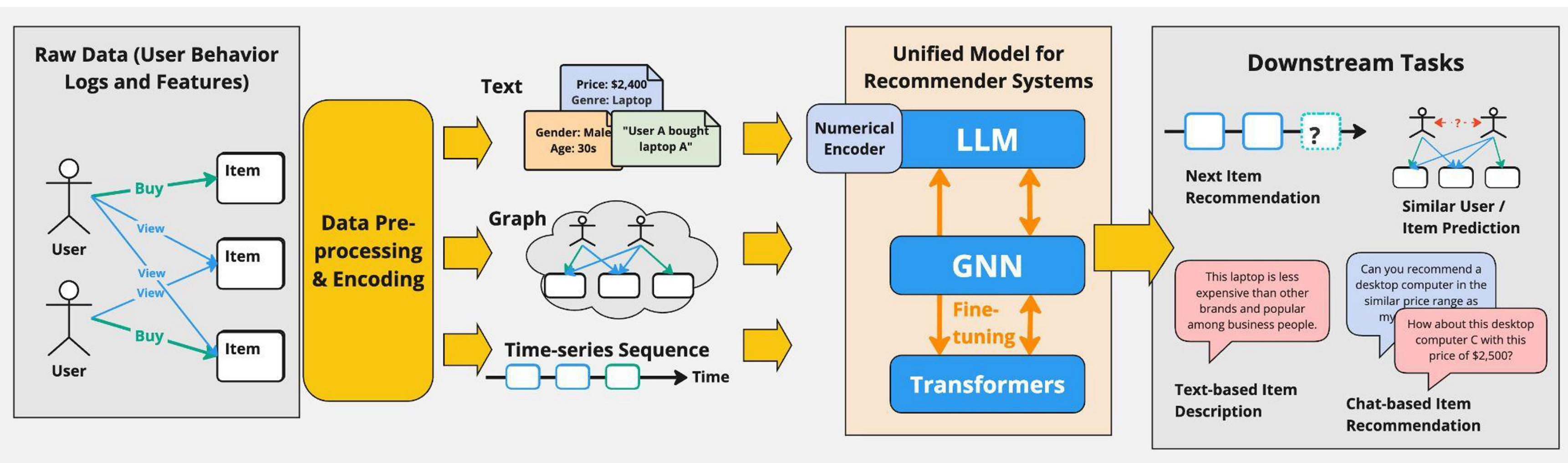
Background:

- In e-commerce and other recommender systems, "tail items"—products purchased by only a small number of users—pose a significant challenge.
- Although auxiliary behavior data (such as browsing and favorites) can provide additional insights, they often introduce irrelevant noise, complicating the accurate modeling of user behavior.

Proposal: Unified Model for Recommender Systems

Multi-behavior Recommendation (MBR) model that integrates LLM, GNN and Transformer models with textual auxiliary behavioral data for more precise user behavior representation.

1. Data Preprocessing & Encoding: Convert user/item attributes and user-item interaction records (browsing and purchases) into different data structures, such as natural language text, graph structures, and time-series sequences.
2. Enhanced LLM: Pretrain LLMs using textual datasets, incorporating knowledge like numerical data encoding optimization or fine-tuning them.
3. Unified Model Training: Train GNN and Transformer models linked with the LLM using graph structures and sequence data that textual information cannot capture, followed by further fine-tuning of the LLMs with outputs.
4. Performance Evaluation with Downstream Tasks: Evaluate our unified model with various downstream tasks, including next-item recommendation, text-based item description, and chat-based item recommendation.



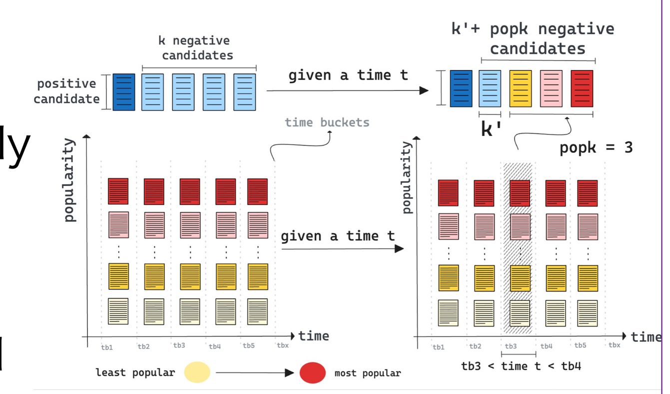
POPK: Enhancing Diversity and Accuracy in News Recommender Systems

Contributions:

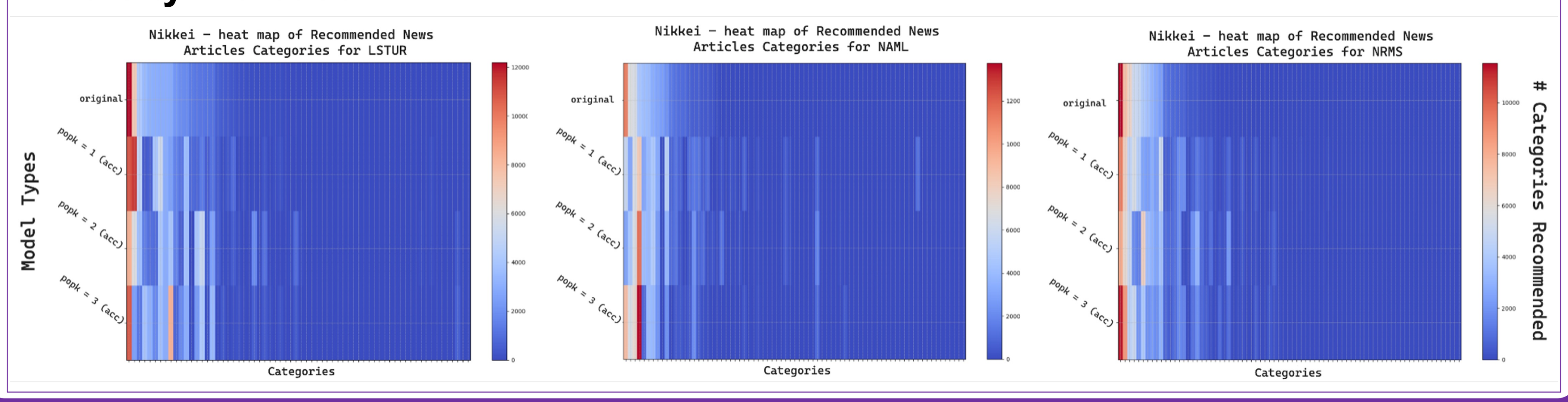
1. A straightforward yet efficient strategy to mitigate the impact of popular news articles, subsequently improving existing methods in terms of both diversity and accuracy;
2. Demonstrations highlighting that *POPK* not only significantly enhances the performance of leading models across various metrics but also offers flexibility for easy adaptation to individual cases;
3. Thorough experiments conducted on real-world data to validate the efficacy of *POPK*.

Overall Idea

- Our proposed method, *POPK*, is based on the idea that popular news articles **always** compete for attention, even if they are not **explicitly** present in the user's impression list. Typically, recommender systems are trained using a negative sampling strategy, where for each positive news article, ***k*** negative news articles are selected from the user's impression list.
- The model might bias toward recommending popular news articles since they are more likely to appear in the candidate sample space.
- The concept of POPK is to *counterfactually* intervene in the selection of these *k* negative items to mitigate bias by incorporating a fixed number (*popk*) of popular news articles.



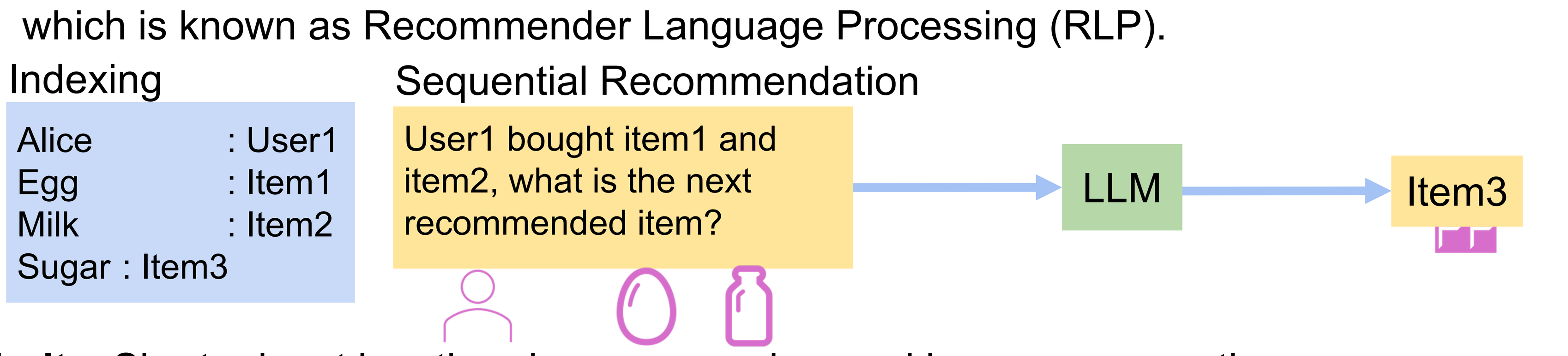
Diversity Increase



Language Processing-based Recommender on Sequential Recommendation via Arithmetic Task

Question: Can the LLM become a recommender?

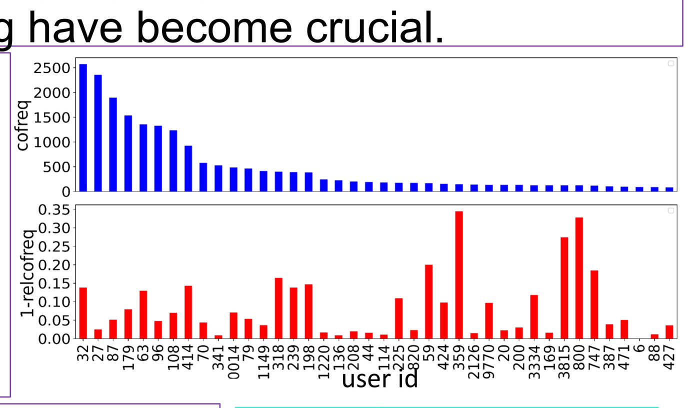
- One approach: Fine-tune LLM and replace the user-item information with index, which is known as Recommender Language Processing (RLP).



Merits: Shorter input length, privacy preserving, and language agnostic.

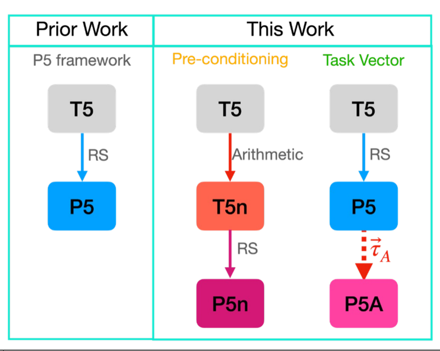
Challenges: Indexing strategy and numerical encoding have become crucial.

Identified Problem: It is hard to capture sequential dependencies when sequential indexing is used, *and* random distribution is applied to numerical encoding. The preliminary result suggests that many adjacent items (like egg and milk!) appear many times in dataset but it is hard to capture this.

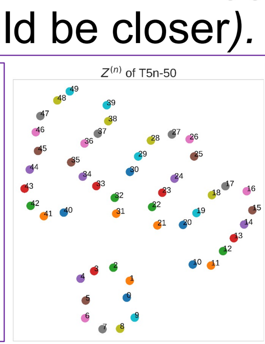


Hypothesis: *Maybe if we make the number embedding become more continuous, we can capture the sequential dependencies among items.* (Model learns that egg, milk, and sugar should be closer*)*.

Proposed Methods:
Model editing,
Using arithmetic task and applies (→):
1) Pre-conditioning
2) Task vector



Why arithmetic? Continuous number embedding.



Improvement? Overall better than baseline (P5, Geng+ 2022 https://arxiv.org/abs/2203.13366) $p<0.05$. (Δ is % improv.)

Model →	Optimized			
Metric → Dataset ↓	hit@5 (Δ)	hit@10(Δ)	ndcg@5(Δ)	ndcg@10(Δ)
Beauty (s:1)	0.0034 (11.9)	0.0039 (0.0)	0.0021 (8.7)	0.0023 (2.7)
Taobao (s:1)	0.1855 (0.9)	0.2188 (-0.1)	0.1413 (0.5)	0.1523 (0.1)
Clothing (s:1)	0.0015 (13.7)	0.0019 (0.0)	0.0009 (19.5)	0.0011 (7.4)
ML100K (s:1)	0.0827 (18.2)	0.1336 (9.6)	0.0507 (18.1)	0.0661 (10.1)
LastFM (s:1)	0.0238 (36.8)	0.0348 (2.7)	0.0135 (29.5)	0.0169 (8.7)
Beauty (u)	0.0032 (12.7)	0.0043 (10.3)	0.0021 (12.9)	0.0025 (10.9)
Taobao (u)	0.1843 (0.0)	0.2199 (0.0)	0.1403 (-0.6)	0.1518 (-0.6)
Clothing (u)	0.0015 (29.8)	0.0021 (13.5)	0.0010 (29.9)	0.0012 (19.3)
ML100K (u)	0.0848 (29.0)	0.1293 (2.5)	0.0504 (19.5)	0.0649 (4.8)
LastFM (u)	0.0238 (13.0)	0.0348 (2.7)	0.0149 (24.3)	0.0174 (8.5)

Conclusion: Arithmetic task can improve language-based recommender via model editing, this is because it helps capturing sequential dependencies which is important in sequential recommendation. Future studies will look into the theoretical analysis.

A Prompting-Based Representation Learning Method for Recommendation with LLMs

Recommender systems have witnessed a transformative shift with the advent of LLMs in the field of Natural Language Processing (NLP). Models such as GPT-3.5/4, Llama, have demonstrated unprecedented capabilities in understanding and generating human-like text that allows for the potential to capture a more profound semantic representation from different contextual information of users and items.

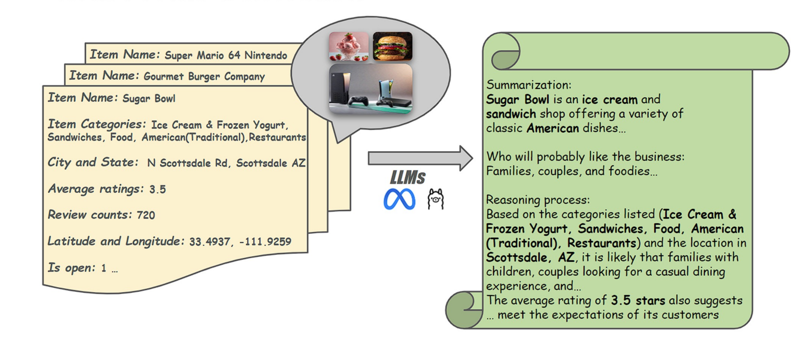
Challenges:

- Leveraging user preferences and item potential consumers through LLMs
- Alignment with the improvement of Recommender Systems

Proposed methods:

Utilize the LLM prompting strategy to create personalized item profiles. These profiles are then transformed into semantic representation spaces using a pre-trained BERT model for text embedding. Furthermore, we incorporate a Graph Convolution Network (GCN) for collaborative filtering representation.

Item Profile Generation



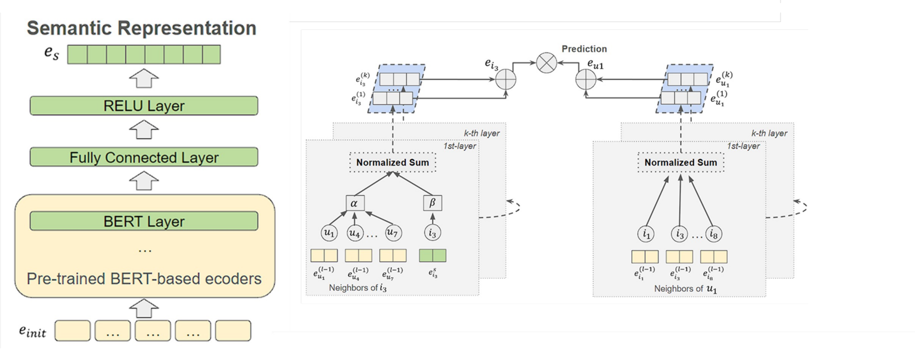


Table 2: The perfomance of the baselines and proposed model P4R

Dataset	Yelp							
Model	Recall@10	Recall@20	NDCG@10	NDCG@20	MRR@10	MRR@20	Hit@10	Hit@20
NGCF	0.1060	0.1565	0.0861	0.1034	0.1374	0.1447	0.2641	0.3686
LightGCN	0.1796	0.2266	0.1615	0.1777	0.2512	0.2572	0.4000	0.4889
SGL	0.1872	0.2536	0.1725	0.1940	0.2750	0.2827	0.4275	0.5386
P4R	0.2030	0.2599	0.1823	0.2009	0.2813	0.2872	0.4444	0.5307
Dataset	**Amazon-VideoGames**							
Model	Recall@10	Recall@20	NDCG@10	NDCG@20	MRR@10	MRR@20	Hit@10	Hit@20
NGCF	0.0639	0.0880	0.0676	0.0754	0.1391	0.1438	0.2322	0.3020
LightGCN	0.1131	0.1449	0.1148	0.1245	0.2153	0.2203	0.3477	0.4213
SGL	0.1340	0.1653	0.1347	0.1442	0.2438	0.2479	0.4023	0.4569
P4R	0.1411	0.1770	0.1569	0.1674	0.2954	0.2988	0.4048	0.4695

Conclusion:

- Our method leverages LLMs to extract features from item descriptions, generating informative and predictive item profiles. By incorporating a GCN-based Collaborative Filtering measure, we demonstrate the effectiveness of aligning two domains for recommendations.
- Various LLMs are to be explored. It is believed that a larger model can produce more detailed profiles, but challenge is the balance between efficiency and accuracy.