Large Language Models for Recommendation: Progresses and Future Direction

Lecture Tutorial For WWW 2024

Organizers: Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, Xiangnan He
Outline

• Introduction (Fuli Feng)
• Background: LM & LM4Rec (Fuli Feng)
• Development of LLMs (Keqin Bao)
• Progress of LLM4Rec
  • LLM4Rec (Keqin Bao & Wenjie Wang)
  • QA & Coffee Break
  • Trustworthy LLM4Rec (Jizhi Zhang)
• Open Problems (Yang Zhang)
• Future Direction & Conclusions (Fuli Feng)
Background of RecSys

Information explosion era

- E-commerce: 12 million items in Amazon.
- Social networks: 2.8 billion users in Facebook.
- Content sharing platforms: 720,000 hours videos uploaded to Youtube per day; 35 million videos posted on TikTok daily

Recommender system

Information seeking via user history feedback

Recommendation
Workflow of Recommender System

1. Train recommender on collected interaction data to capture user preferences.
2. Recommender generates recommendations based on estimated preferences.
3. User engages with the recommended items, forming new data, affected by open world.
4. Train recommender with new data again, either refining user interests or capturing new ones.
Background of RecSys

- Core idea of personalized recommendation
  - Collaborative filtering (CF):
    Making automatic predictions (filtering) about the interests of a user by collecting preferences from many users (collaborating).

Images from: Neural Collaborative Filtering,
Core idea of personalized recommendation

- Collaborative filtering (CF):
  Making automatic predictions (filtering) about the interests of a user by collecting preferences from many users (collaborating).

Memory-based CF
- User CF
- Item CF

Model-based CF
- MF
- FISM
- ...

Neural CF

GCN-based CF

Xiangnan He et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation
Core idea of personalized recommendation

- Collaborative filtering (CF): collaborative information
- Content/context-aware models (CTR models): side information + context information
  - Click-Through Rate (CTR) prediction

Factorization machines: FM, NFM, DeepFM

Neural network: DIN, AutoInt
The development of LMs

NLP 1.0: Dictionary/Vocabulary + Rules

Before 1990

NLP 2.0: Statistical NLP

1990

NLP 3.0: Deep Learning for NLP

2010

NLP 4.0: Pretrained Language Models

2018

LLMs: ChatGPT

2022

GPT4、Gemini1.5、LLaMA3、

GLM4、Qwen1.5

Large Language Model: billions of parameters, emergent capabilities

• Rich knowledge & Language Capabilities
• Instruction following
• In-context learning
• Chain-of-thought
• Planning
• …
The development of LMs

- LLMs such as ChatGPT and GPT4 have influenced many fields in CS and IT industry
  - They have eliminated a wide range of research in basic NLP and conversational system, etc.

Recommender System + LLMs?
Outline

• Introduction
• **Background: LM & LM4Rec**
• Development of LLMs
• Progress of LLM4Rec
• Open Problems
• Future Direction & Conclusions
Development of LMs

Transformer

Encoder

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other queries, keys, values are computed from encoder states

Decoder

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Decoder-encoder attention: target token looks at the source queries – from decoder states; keys and values from encoder states

Decoder self-attention (masked): tokens look at the previous tokens queries, keys, values are computed from decoder states

Bert: pre-training of deep bidirectional transformers

Mask Language Modeling, bi-direction

Encoder (advantage) --> understanding
Development of LMs

- GPT2: generative pre-trained transformer
- Causal language modeling
- Decoder (advantage) -- Generation
- unsupervised multi-task learner

How can recommender systems benefit from LMs

- **Model architecture:** Transformer, Self-attention

- **Representation:**
  - Textual feature,
  - item representation,
  - knowledge representation

- **Task formulation**
  Use language to formulate the recommendation task

- **Learning paradigm:**
  Pretrain-finetune, Prompt learning
Overview of LM4rec

- LMs for recommendation
  - Utilizing LMs' model structure for recommendation.
    - ID-based: BERT4Rec, SASRec ...
    - Text-based: Recformer ...
  - LM as item encoder. UniSRec, VQRec, MoRec ...
  - Recommendation as natural language processing.
    - ID-based: P5, VIP5 ...
    - Text-based: M6-Rec, Prompt4NR ...
Utilizing LM Model Structure

Bert4Rec: ID-based

Natural Language:
• Token sequence
• Inter-token correlations

RecSys:
• ID sequence
• Inter-item correlations

Training recommender by masked item prediction as BERT.

Utilizing LM Model Structure

- **Recformer: text-based**
  - Text is all you need (NO item ID)
    - Only use texts to represent items.
    - Low resource, better cold-start recommendation.

Utilizing LM Model Structure

- **Recformer**: text-based
- Text is all you need (NO item ID)

![Diagram of Recformer Model Structure]

(a) Recformer Model Structure

(b) Pretraining

UniSRec

Enhance the recommendation model by using LMs to encode the natural language representation of items.

Figure 1: The overall framework of the proposed universal sequence representation learning approach (UniSRec).
Recommendation as NLP

- **P5: use natural language to describe different rec. tasks.**

  - **Multi-task prompts**
    - Sequential recommendation
    - Rating prediction
    - Explain generation
    - Review summarization
    - Direct recommendation

  - Sequential Recommendation
    - I find the purchase history list of user_15466: 4110 -> 4467 -> 4468 -> 4472. I wonder what is the next item to recommend to the user. Can you help me decide?

  - Rating Prediction
    - What star rating do you think user_23 will give item_73917?

  - Explanation Generation
    - Help Hong "Old boy" generate a 5-star explanation about this product: OtterBox Defender Case for iPhone 3G, 3G S (Black) (Retail Packaging)

  - Review Summarization
    - Give a short sentence describing the following product review from Mom of 3 yo girl: First it came with the packaging open and then as soon as my son took it out it was so easily broken. Hopefully a little glue will fix it.

  - Direct Recommendation
    - Pick the most suitable item from the following list and recommend to user_250: 
      - 4915, 1823, 3112, 3821, 3773, 520, 7384, 7469, 9318, 3876, 1143, 789, 395, 3824, 3587, 10396, 2766, 7498, 2490, 3232, 9711, 2075, 1427, 9923, 3097, 3594, 6469, 9460, 6956, 9154

Recommendation as NLP

- **P5 Architecture:**
  - Autoregressive decoding
  - Users and items are represented with ID information

---

M6-Rec: represent users/item with plain texts and converting the tasks to either language understanding or generation

- Understanding (scoring) task: CTR, CVR prediction
- Generation task: personalized product design, explanation generation…

User description

[BOS’] December. Beijing, China. Cold weather. A male user in early twenties, searched “winter stuff” 23 minutes ago, clicked a product of category “jacket” named “men’s lightweight warm winter hooded jacket” 19 minutes ago, clicked a product of category “sweatshirt” named “men’s plus size sweatshirt stretchy pullover hoodies” 13 minutes ago, clicked … [EOS’]

[BOS] The user is now recommended a product of category “boots” named “waterproof hiking shoes mens outdoor”. The product has a high population-level CTR in the past 14 days, among the top 5%. The user clicked the category 4 times in the last 2 years. [EOS]

Item description
Outline

- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- Open Problems
- Future Direction & Conclusions
Developments of LLMs

- **Scaling Laws**
  - The greater the amount of the data and the model parameters, the better the performance of the model.
  - Performance can be predicted.

Jared Kaplan et al. Scaling Laws for Neural Language Models arxiv
Developments of LLMs

- **Scaling Laws**
  - The greater the amount of the data and the model parameters, the better the performance of the model
  - Performance can be predicted
Developments of LLMs

Scaling Laws

Larger models require fewer samples to reach the same performance

The optimal model size grows smoothly with the loss target and compute budget

Jared Kaplan et al. Scaling Laws for Neural Language Models arxiv
Developments of LLMs

Align with human

Step 1
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Reinforcement Learning from Human Feedback (RLHF)

- “Write me a poem about the history of jazz”
  - Reward model
  - LM policy
  - label rewards
  - sample completions
  - maximum likelihood
  - reinforcement learning

Direct Preference Optimization (DPO)

- “Write me a poem about the history of jazz”
  - Preference data
  - maximum likelihood
  - final LM

Long Ouyang et al., 2022 Training language models to follow instructions with human feedback NeurIPS 2022
Rafael Rafailov et al., 2023 Direct Preference Optimization: Your Language Model is Secretly a Reward Model NeurIPS 2023
Developments of LLMs

More and more LLMs have shown powerful capabilities
Developments of LLMs

- Multi-model to Multi-model unified model is now developing at a rapid pace.
Augmented capabilities of LLMs

- Emergent abilities of LLM
  - Sufficient world knowledge
  - Chatting
  - Incontext Learning & Instruction Following
  - Reasoning & Planning
  - Tool using
  - LLM as an Agent
  - ...

Augmented capabilities of LLMs

- In-context Learning
  - Following their example to override the semantic prior

---

**Regular ICL**

- Natural language targets: (Positive/Negative) sentiment
  - Contains no wit [...] \n  - Very good viewing [...] \n  - A smile on your face \n  - Language Model
    - Positive

**Flipped-Label ICL**

- Flipped natural language targets: (Negative/Positive) sentiment
  - Contains no wit [...] \n  - Very good viewing [...] \n  - A smile on your face \n  - Language Model
    - Negative

**SUL-ICL**

- Semantically-unrelated targets: {Foo/Bar}, {Apple/Orange}, {A/B}
  - Contains no wit [...] \n  - Very good viewing [...] \n  - A smile on your face \n  - Language Model
    - Foo

*Jerrt Wei et al. Language Models Do In-context Learning Differently*
Augmented capabilities of LLMs

☐ Instruction following

Q. what is vantage software

- Official sources
- Discussion forum
- Easy-to-understand visual materials

... prefer formal sources with in-depth analysis and reviews from industry professionals.

... find opinions and personal experiences from other tech enthusiasts.

My teacher assigns homework... easy to understand and many examples.

Jerrt Wei et al. Language Models Do In-context Learning Differently
Augmented capabilities of LLMs

- **Reasoning & Planning**
  - LLM can decompose the problem into simple sub-problems to improve their ability

<table>
<thead>
<tr>
<th>Standard Prompting</th>
<th>Chain-of-Thought Prompting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Input</strong></td>
<td>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</td>
</tr>
<tr>
<td></td>
<td>A: The answer is 11.</td>
</tr>
<tr>
<td>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</td>
<td></td>
</tr>
<tr>
<td><strong>Model Output</strong></td>
<td>A: The answer is 27. ✗</td>
</tr>
<tr>
<td></td>
<td>Q: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.</td>
</tr>
<tr>
<td></td>
<td>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.</td>
</tr>
<tr>
<td></td>
<td>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</td>
</tr>
<tr>
<td></td>
<td>A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓</td>
</tr>
</tbody>
</table>

*Jason Wei et al, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*
Augmented capabilities of LLMs

- Reasoning & Planning
  - LLM can break down the target task according to the environment and develop a plan.

Yao et al., REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS
Augmented capabilities of LLMs

- LLM as an Agent

Images from https://lilianweng.github.io/posts/2023-06-23-agent/
Multi-Agent

- Group intelligence surpasses individual intelligence
- Cooperative for complementary / Adversarial for advancement

**Cooperative Engagement**

- Disordered
  - To create a product, we should...
  - Let’s…
  - I think the first step is...
  - Firstly, we should...

- Ordered
  - Manager
    - The theme of our product is...
  - Designer
    - The architecture of the product is...
  - Engineer
    - Programming ...
    - def main():
  - Tester
    - The product has the following issues: ...

**Adversarial Interactions**

- Designer
  - I think users need a simplified interface.
- Engineer
  - Good idea, but... technical limitations might affect performance.
- Designer
  - True... while simplification does enhance user experience.
- Engineer
  - Yeah, but performance issues also impact overall satisfaction. I will try my best to balance both aspects.

Zhiheng Xi et al, The Rise and Potential of Large Language Model Based Agents: A Survey
## LLMs for Recommendation

### How recommender systems benefit from LLMs

<table>
<thead>
<tr>
<th>Representation:</th>
<th>Interaction:</th>
<th>Generalization:</th>
<th>Generation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textual feature,</td>
<td>Acquire user information needs via dialog (chat)</td>
<td>cross-domain, knowledge compositional-generalization</td>
<td>Personalized content generation, explanation generation</td>
</tr>
<tr>
<td>item representation,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>knowledge representation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Learning paradigm:** Pretrain-finetune, Instruction-tuning, Preference-alignment
- **Model architecture:** Transformer, Self-attention,
LLMs for Recommendation

- Key Challenge
  - Mismatch between pretraining objective and recommendation
  - Tend to rely on semantics, and another important aspect of recommendation tasks is collaborative information.
Outline

• Introduction
• Background: LM & LM4Rec
• The progress of LLM4Rec
  • Development of LLMs
  • LLMs for Recommendation
    • ICL
    • Tuning
    • Agent
• Open Problems
• Conclusions
Progress of LLM4Rec

Three dimensions:

Metrics
From accuracy to trustworthiness
such as privacy, fairness, etc.

Information modalities
From the text modality to multiple modalities

How to utilize LLMs
In-context learning
Tuning
Agent
Progress of LLM4Rec

Three dimensions:

- **Information modalities**: From the *text modality* to multiple modalities
- **Metrics**: From accuracy to *trustworthiness* such as privacy, fairness, etc.
- **How to utilize LLMs**: In-context learning, Tuning, Agent
In-context learning

- LLMs has rich world knowledge, wonderful abilities like reasoning, instruction following, in-context learning.
- The LLMs itself could be leveraged for recommendation by in context learning.
- Existing works on in-context learning:
  - Ask LLM for recommendation
  - Serving as knowledge augmentation for traditional recsys
  - Optimize the prompt used for recommendation
  - Directly used for conversational recommender system
In-context learning: directly ask LLMs for recommendation

- Prompt construction

Three different ways of measuring ranking abilities:

\[
\hat{y}_i = LLM_{\text{point}} (I, D, f(h', c' | u))
\]

\[
\hat{y}_{i m > i n} = LLM_{\text{pair}} (I, D, f(h', c' | u))
\]

\[
\hat{y}_{i 1}, \hat{y}_{i 2}, \ldots, \hat{y}_{i k} = LLM_{\text{list}} (I, D, f(h', c' | u))
\]

Figure 1: The overall evaluation framework of LLMs for recommendation. The left part demonstrates examples of how prompts are constructed to elicit each of the three ranking capabilities. The right part outlines the process of employing LLMs to perform different ranking tasks and conduct evaluations.
In-context learning: re-ranking given candidated items

Task formulation:

- Using historical interaction to rank items retrieved by existing recsys.
- Input: language instructions created with historical interactions and candidate items
- Output: ranking of the candidate items
In-context learning: ranking given candidated items

- **Tree types of prompts:**
  - **Sequential prompting:** describing History using language
    "I’ve watched the following movies in the past in order: ‘0. Multiplicity’, ‘1. Jurassic Park’, . . .."
  - **Recency-focused prompting:** *emphasize most recent interactions*
    “I’ve watched the following movies in the past in order: ‘0. Multiplicity’, ‘1. Jurassic Park’, . . .. Note that my most recently watched movie is Dead Presidents. . . .”
  - **In-context learning (ICL):** *providing recommendation example*
    “If I’ve watched the following movies in the past in order: ‘0. Multiplicity’, ‘1. Jurassic Park’, . . ., then you should recommend Dead Presidents to me and now that I’ve watched Dead Presidents, then . . .”
In-context learning: knowledge enhancement

Traditional RecSys vs ICL-based RecSys

Traditional RecSys

- Inference fast but being closed system, generating recommendations relying on local dataset

Directly ask LLMs for recommendation

- Could leverage open-world knowledge, but:
  1) not trained on specific recommendation task
  2) Inference slowly
  3) hard to correctly answer compositional questions

Extract and inject LLM's world knowledge into traditional recommender system

ICL: KAR

In-context learning: knowledge enhancement

Obtain knowledge beyond local rec dataset:
1) Generate reasoning knowledge on user preference (factors affect preference)
2) Generate factual knowledge about items

Knowledge Adaption Stage
encode the textual knowledge and mapping it into recommendation space

Knowledge Utilization
Use the knowledge obtained from LLMs as additional features

ICL: LLM4Rec

- Implicit Feedback Augmentor
  - Implicit Feedback: \( PU_u \)
  - User Profiling & Item Attribute Enhancing
    - \( LLM(\hat{P}_u) \)
    - Augmented Training Data
  - User Profiling
  - Item Attribute Enhancing
  - Augmented Attribute

1) Augmenting user-item interactions
2) Enhancing item attributes
3) User profiling

- Recommend user with movies based on user history that each movie with title, year, genre.
  - History:
    - Heart and Souls (1993), Comedy|Fantasy
    - Men with Brooms (2002), Comedy|Drama|Romance
  - Please output the following information of user, output format: \{age: , gender: , liked genre: , disliked genre: , liked directors: , country: , language: \}

- Generate user profile based on the history of user, that each movie with title, year, genre.
  - History:
    - Heart and Souls (1993), Comedy|Fantasy
    - Men with Brooms (2002), Comedy|Drama|Romance
  - Example profile:

- User Profile
  - Provide the inquiry information of a given movie.
  - The inquiry information is: director, country, language. And please output them in the form of:
    - director, country, language

- Item Attribute
  - Ron Underwood, USA, English
ICL: Automatically adjust and optimize prompts for recommendation

Zhu Sun et al. Large Language Models for Intent-Driven Session Recommendations
ICL: KECRS

- ICL for conversational recommender system
- Users chat with chatbot with natural language
- Chatbot analyses user interest
- Chatbot provide recommendation

KECRS: Towards Knowledge-Enriched Conversational Recommendation System
ICL for conversational recommender system

- Input: task description $T$, format requirement $F$ and conversation context $S$
- LLMs analyz the input data
- LLMs generate the recommendation list


ICL: LLMCRS
Progress of LLM4Rec

Three dimensions:

Metrics
From accuracy to trustworthiness
such as privacy, fairness, etc.

Information modalities
From the text modality to multiple modalities

How to utilize LLMs
In-context learning
Tuning
Agent
In-context learning is not enough. In complex scenarios, ChatGPT usually gives positive ratings or refuse to answer.

Need to **align** LLM with recommendation task!

Keqin Bao et al. Recsys, TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. 2023
Motivation: lack of recommendation task tuning in LLM pre-training

→ tune LLMs with the recommendation data to align with the recommendation task

Existing work on tuning LLMs for recommendation:

**Discriminative manner**

- Following traditional rec task, provide candidates:
  - pointwise, pairwise, listwise

- **PEFT tuning**
  - TALLRec [1]
  - LLM-TRSR [5]
  - LLamaRec [4]
  - GLRec [8] ......

- **Full tuning**
  - InstructRec [2]
  - LLMunderPre [3]
  - ......

**Generative manner**

- Following the pretraining task, do not provide candidates:
  - directly generate items

- **BigRec** [6]
- **TransRec** [7]
- **LC-Rec** [10]
- **GIRL** [9] ......

Tuning LLM4Rec: TALLRec

- **TALLRec: Instruction-tuning**

  - Fine-tune 4M parameters by few-shot samples via the generative loss
  - Quickly adapt to new tasks

  ![Diagram of TALLRec Framework](image)

  - Lightweight Tuning
    - LoRA
    - Input
    - LLM
    - Output
  - LLM4Rec Framework
    - LLM
    - Alpaca Tuning
    - Rec-Tuning
  - Rec-Tuning Samples
    - Instruction Input
    - Output
    - \( \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{y} \log (P_{\theta+\theta}(y_t | x, y_{<t})) \)

  - Like or not
    - LLM with LoRA
    - User features + item features
      - Use item titles as the input
      - Better for cold-start recommendation

Performance significantly improves by fine-tuning few-shot samples.

Keqin Bao et al. Recsys, TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. 2023
**TALLRec: Cross-domain generalization**

- Learning from movie scenario can directly recommend on books, and vice versa.
- LLM can leverage domain knowledge to accomplish recommendation tasks after acquiring the ability to recommend.

Keqin Bao et al. Recsys, TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. 2023
Tuning LLM4Rec: LLM-TRSR

- Text-Rich Sequential Recommendation

  - LLM for preference summary
    - Hierarchical summarization
    - Recurrent summarization
    - Supervised fine-tuning
  - Given user preference summary, recently interacted items, and candidate items, LLMs are tuned for recommendation

---

InstructRec

- User could express their needs diversely: vague or specific; implicit or explicit
- LLM should understand and follow different instructions for recommendation

---

Junjie Zhang et al. Arxiv, Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. 2023
InstructRec: Instruction construction:

- **Format:** Preference: none/implicit/explicit  Intention: none/vague/specific  task: pointwise/pairwise/listwise

<table>
<thead>
<tr>
<th>Instantiation</th>
<th>Model Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_1, I_0, T_0)</td>
<td>The user has purchased these items: &lt;historical interactions&gt; . Based on this information, is it likely that the user will interact with &lt;target item&gt; next?</td>
</tr>
<tr>
<td>(P_2, I_0, T_1)</td>
<td>You are a search engine and you meet a user’s query: &lt;explicit preference&gt; . Please respond to this user by selecting items from the candidates: &lt;candidate items&gt;.</td>
</tr>
<tr>
<td>(P_3, I_1, T_2)</td>
<td>As a recommender system, your task is to recommend an item that is related to the user’s &lt;vague intention&gt; . Please provide your recommendation.</td>
</tr>
<tr>
<td>(P_4, I_2, T_3)</td>
<td>Suppose you are a search engine, now the user search that &lt;specific Intention&gt; , can you generate the item to respond to user’s query?</td>
</tr>
<tr>
<td>(P_5, I_0, T_4)</td>
<td>Here is the historical interactions of a user: &lt;historical interactions&gt; . His preferences are as follows: &lt;explicit preference&gt; . Please provide recommendations .</td>
</tr>
<tr>
<td>(P_6, I_1, T_5)</td>
<td>The user has interacted with the following &lt;historical interactions&gt; . Now the user search for &lt;vague intention&gt; , please generate products that match his intent.</td>
</tr>
<tr>
<td>(P_7, I_2, T_6)</td>
<td>The user has recently purchased the following &lt;historical items&gt; . The user has expressed a desire for &lt;specific intention&gt; . Please provide recommendations.</td>
</tr>
</tbody>
</table>

- **Instruction generation:** #1 using ChatGPT to generate user preferences and intentions based on interactions

- **Interaction:**
  - Explicit preference

- **Generated Explicit Preference:**
  - [Raw Behavior Sequence]: “1. Resident Evil: Revelations 2 - PS 4 $ightarrow$ 2. Resident Evil 4 - PS 4.”
  - [Generated Explicit Preference]: “He prefers horror-based games with a strong narrative.”

- **Review:**
  - Vague intention

- **Generated Vague Intention:**
  - [Raw Target Review]: “My son loves ... of the game. I’m happy I bought this for him.”
  - [Generated Vague Intention]: “I enjoy buying games for my son that he enjoys.”

#2 Increasing the instruction diversity via multiple strategies such as CoT

Junjie Zhang et al. Arxiv, Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. 2023
# Tuning LLM4Rec: InstructRec

## InstructRec

- Instruction construction
  - Quality: human evaluation

## Quality Review Question

<table>
<thead>
<tr>
<th>Quality Review Question</th>
<th>Preference</th>
<th>Intention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the instruction generated from the user's related information?</td>
<td>93%</td>
<td>90%</td>
</tr>
<tr>
<td>Does the teacher-LLM provide related world knowledge?</td>
<td>87%</td>
<td>22%</td>
</tr>
<tr>
<td>Does the instruction reflect the user’s preference/ intention?</td>
<td>88%</td>
<td>69%</td>
</tr>
<tr>
<td>Is the instruction related to target item?</td>
<td>48%</td>
<td>69%</td>
</tr>
</tbody>
</table>

## Instruction tuning:

- Supervised fine-tuning, tuning all model parameters (3B Flan-T5-XL)

$$
\mathcal{L} = \sum_{k=1}^{B} \sum_{j=1}^{Y_k} \log P \left( Y_{k,j} \mid Y_{k,<j}, I_k \right),
$$

where $Y_k$ is the desired system responses for the $k$-th instance, $I_k$ is the instruction of the $k$-th instance, and $B$ is the batch size.
Motivation: lack of recommendation task tuning in LLM pre-training

→ tune LLMs with the recommendation data to align with the recommendation task

Existing work on tuning LLMs for recommendation:

**Discriminative manner**

- Following *traditional rec task*, provide candidates: pointwise, pairwise, listwise

**Generative manner**

- Following *the pretraining task*, do not provide candidates: directly generate items

**PEFT tuning**

- TALLRec [1]
- LLM-TRSR [5]
- LLamaRec [4]
- GLRec [8] ......

**Full tuning**

- InstructRec[2]
- LLMunderPre[3]
- ......

**BigRec [6]**

- TransRec [7]
- LC-Rec [10]
- GIRL[9] ......

Tuning LLM4Rec: BIGRec

- **BIGRec**
  - **Generation + Grounding**
    - Given user interaction history in natural language, LLMs aim to generate the next item as recommendation.
    - However, LLMs do not know how to represent an item via token sequence in the recommendation scenario.
    - Besides, the item generated by the LLM may not exist in the actual world.

**Grounding Paradigm**

- **Language Space**
  - *Step 1: instruction tuning*
- **Recommendation Space**
  - *Step 2: L2 distance grounding*
- **Actual Item Space**

Tuning LLM4Rec: BIGRec

- **BIGRec**
  - Few-shot tuning

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>NG@1</th>
<th>NG@3</th>
<th>NG@5</th>
<th>NG@10</th>
<th>NG@20</th>
<th>HR@1</th>
<th>HR@3</th>
<th>HR@5</th>
<th>HR@10</th>
<th>HR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>GRU4Rec</td>
<td>0.0015</td>
<td>0.0034</td>
<td>0.0047</td>
<td>0.0070</td>
<td>0.0104</td>
<td>0.0015</td>
<td>0.0047</td>
<td>0.0079</td>
<td>0.0147</td>
<td>0.0281</td>
</tr>
<tr>
<td></td>
<td>Caseer</td>
<td>0.0020</td>
<td>0.0035</td>
<td>0.0052</td>
<td>0.0078</td>
<td>0.0109</td>
<td>0.0020</td>
<td>0.0046</td>
<td>0.0088</td>
<td>0.0171</td>
<td>0.0293</td>
</tr>
<tr>
<td></td>
<td>SASRec</td>
<td>0.0023</td>
<td>0.0051</td>
<td>0.0062</td>
<td>0.0082</td>
<td>0.0117</td>
<td>0.0023</td>
<td>0.0070</td>
<td>0.0097</td>
<td>0.0161</td>
<td>0.0301</td>
</tr>
<tr>
<td></td>
<td>P5</td>
<td>0.0014</td>
<td>0.0026</td>
<td>0.0036</td>
<td>0.0051</td>
<td>0.0069</td>
<td>0.0014</td>
<td>0.0035</td>
<td>0.0059</td>
<td>0.0107</td>
<td>0.0176</td>
</tr>
<tr>
<td></td>
<td>DROS</td>
<td>0.0022</td>
<td>0.0040</td>
<td>0.0052</td>
<td>0.0081</td>
<td>0.0112</td>
<td>0.0022</td>
<td>0.0051</td>
<td>0.0081</td>
<td>0.0173</td>
<td>0.0297</td>
</tr>
<tr>
<td></td>
<td>GPT4Rec-LLaMA</td>
<td>0.0016</td>
<td>0.0022</td>
<td>0.0024</td>
<td>0.0028</td>
<td>0.0035</td>
<td>0.0016</td>
<td>0.0026</td>
<td>0.0030</td>
<td>0.0044</td>
<td>0.0074</td>
</tr>
<tr>
<td>BIGRec (1024)</td>
<td>0.0176</td>
<td>0.0214</td>
<td>0.0230</td>
<td>0.0257</td>
<td>0.0283</td>
<td>0.0176</td>
<td>0.0241</td>
<td>0.0281</td>
<td>0.0366</td>
<td>0.0471</td>
<td></td>
</tr>
<tr>
<td>Improve</td>
<td>654.29%</td>
<td>323.31%</td>
<td>273.70%</td>
<td>213.71%</td>
<td>142.55%</td>
<td>654.29%</td>
<td>244.71%</td>
<td>188.39%</td>
<td>111.97%</td>
<td>56.55%</td>
<td></td>
</tr>
</tbody>
</table>

| Game    | GRU4Rec   | 0.0013| 0.0016| 0.0018| 0.0024| 0.0030| 0.0013| 0.0018| 0.0024| 0.0041| 0.0069|
|         | Caseer    | 0.0007| 0.0012| 0.0019| 0.0024| 0.0035| 0.0007| 0.0016| 0.0032| 0.0048| 0.0092|
|         | SASRec    | 0.0009| 0.0012| 0.0015| 0.0020| 0.0025| 0.0009| 0.0015| 0.0021| 0.0037| 0.0057|
|         | P5        | 0.0002| 0.0005| 0.0007| 0.0010| 0.0017| 0.0002| 0.0007| 0.0012| 0.0023| 0.0049|
|         | DROS      | 0.0006| 0.0011| 0.0013| 0.0016| 0.0022| 0.0006| 0.0015| 0.0019| 0.0027| 0.0052|
|         | GPT4Rec-LLaMA | 0.0000 | 0.0000 | 0.0000 | 0.0001 | 0.0001 | 0.0000 | 0.0000 | 0.0000 | 0.0002 | 0.0002 |
| BIGRec (1024) | 0.0133 | 0.0169 | 0.0189 | 0.0216 | 0.0248 | 0.0133 | 0.0195 | 0.0243 | 0.0329 | 0.0457 |
| Improve | 952.63%   | 976.26% | 888.19% | 799.64% | 613.76% | 952.63% | 985.19% | 660.42% | 586.11% | 397.10% |

- BIGRec significantly surpasses baselines by few-shot tuning.
- Improvement of BIGRec is significantly higher on Game compared to on Movie.
  - possibly due to the varying properties of popularity bias between the two datasets.

Tuning LLM4Rec: BIGRec

- **BIGRec**

**Quickly adapt to recommendation!**

**Performance w.r.t. NDCG@20**

- **Relative improvements w.r.t. NDCG@20**

Not proficient in utilizing collaborative filtering signals in interactions!


- **BIGRec**
  - Injecting statistical information into BIGRec at Step 2: L2 distance grounding
  
  ![Graphs showing NDCG@K and HR@K improvements](image)

  - By incorporating popularity, BIGRec achieves significant improvements \( w.r.t. \) NDCG@\( K \) and HR@\( K \), particularly for a larger \( K \).

  - Incorporating collaborative information into BIGRec yields more significant enhancements than conventional models.
LLM for generative recommendation

- Two key problems of LLM4Rec
  - Item tokenization: index items into language space
  - Item generation: generate items as recommendations
Tuning LLM4Rec: TransRec

- **Item indexing: multi-facet identifier**
  - ID 15826  (Distinctiveness)
  - Title Wilson Indoor Basketballs  (Semantics)
  - Attribute Sports

- **Generation grounding:**
  - Position-free constrained generation
  - FM-index: special prefix tree that supports search from any position of the identifier corpus.

- **Instruction data reconstruction**

  **Instruction Input**
  - ID
    - Given the following purchase history of a user, what is the next possible item to be purchased by the user?
      - 15826; 8792; 513; 7382; 9014; ID
  - Title
    - Given the following purchase history of a user, what is the next possible item to be purchased by the user?
      - Wilson NBA Basketballs; Advancourt Sneakers; ...
      - Logitech K270 Wireless Keyboard; ID
  - Attribute
    - Given the following attributes of purchase history of a user, what is the next possible attribute of item to be purchased by the user?
      - Sports; Shoe; Headphone & Earphones; ...
      - Electronics; ID

  **Instruction Output**
  - 23
  - Wireless Mouse
  - Electronics

  **Constrained & Position-free Generation**
  - LLMs
  - FM-index

  **Generated Identifiers**
  - 95, 7002, 3865, 3789, 6055, ...
  - mouse pad, monitor, cables, wireless ...
  - electronics, IT, accessories, ...

  **Aggregated Grounding**
  - ID: 95
  - Title: LG monitor
  - Type C cables
  - Attribute: Tech accessories ...

User’s historical interactions in three facets

Tuning LLM4Rec: TransRec

- **Strong generalization ability**
  - Item group analysis
    - From warm to cold items
  - User group analysis
    - From dense users to sparse users

<table>
<thead>
<tr>
<th>N-shot</th>
<th>Model</th>
<th>Warm</th>
<th>Cold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@5</td>
<td>N@5</td>
</tr>
<tr>
<td>1024</td>
<td>LightGCN</td>
<td>0.0205</td>
<td>0.0125</td>
</tr>
<tr>
<td></td>
<td>ACAVE</td>
<td>0.0098</td>
<td>0.0057</td>
</tr>
<tr>
<td></td>
<td>P5</td>
<td>0.0640</td>
<td>0.0016</td>
</tr>
<tr>
<td></td>
<td>TransRec-B</td>
<td>0.0039</td>
<td>0.0024</td>
</tr>
<tr>
<td></td>
<td>TransRec-L</td>
<td>0.0141</td>
<td>0.0070</td>
</tr>
<tr>
<td>2048</td>
<td>LightGCN</td>
<td>0.0186</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td>ACAVE</td>
<td>0.0229</td>
<td>0.0136</td>
</tr>
<tr>
<td></td>
<td>P5</td>
<td>0.0047</td>
<td>0.0030</td>
</tr>
<tr>
<td></td>
<td>TransRec-B</td>
<td>0.0052</td>
<td>0.0027</td>
</tr>
<tr>
<td></td>
<td>TransRec-L</td>
<td>0.0194</td>
<td>0.0126</td>
</tr>
</tbody>
</table>

* The bold results highlight the superior performance compared to the best LLM-based recommender baseline.

- On the item side, TransRec-L with LLMs has remarkable generalization ability with vase knowledge base, especially on cold-start recommendation under limited data.
- On the user side, TransRec significantly **improves the performance of sparse users** with fewer interactions.
Tuning LLM4Rec: LC-Rec

- **LC-Rec**
  - Item indexing: utilize Residual-Quantized Variational AutoEncoder (RQ-VAE) to encode item semantic information as identifiers.
  - Multiple alignment tasks to inject collaborative signals
Progress of LLM4Rec

Three dimensions:

Information modalities
From the text modality to multiple modalities

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.

How to utilize LLMs
In-context learning
Tuning
Agent
LLM-empowered Agents for Recommendation

Agent as User Simulator

- **Main idea:** using agents to simulate user behavior for real-world recommendation.

  - RecAgent\(^1\), Agent4Rec\(^2\)

Agent for Recommendation

- **Main idea:** harnessing the powerful capabilities of LLMs, such as reasoning, reflection, planning and tool usage, for recommendation.

  - RecMind\(^3\), InteRecAgent\(^4\), BiLLP\(^5\), Multi-Agent Collaboration\(^6\)

---


\(^5\) Wentao Shi et al. 2023. Large Language Models are Learnable Planners for Long-Term Recommendation. in SIGIR 2024.

\(^6\) Jiabao Fang et al. A Multi-Agent Conversational Recommender System. Arxiv 2024
Agent: RecAgent

- **LLM-based agent for user simulation**

  - User simulation is a fundamental problem in human-centered applications.
  - Traditional methods **struggle to simulate** complex user behaviors.
  - LLMs show potential in human-level intelligence and generalization capabilities.

Agent: RecAgent

- **Recommendation Behaviors**
  Agent chooses to search or receive recommendations, selects movies, and stores feelings after watching.

- **Chatting Behaviors**
  Two agents discuss and stored the conversation in their memories.

- **Broadcasting Behaviors**
  An agent posts a message on social media, received by friends and stored in their memories.
Agent: Agent4Rec

- Agent4Rec, a simulator with 1,000 LLM-empowered generative agents.

- Agents are trained by the MovieLens-1M dataset, embodying varied social traits and preferences.

- Each agent interacts with personalized movie recommendations in a page-by-page manner and undertakes various actions such as watching, rating, evaluating, exiting, and interviewing.

To what extent can LLM-empowered generative agents truly simulate the behavior of genuine, independent humans in recommender systems?

User Taste Alignment

Table 1: User taste discrimination.

<table>
<thead>
<tr>
<th>1:m</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1</td>
<td><strong>0.6912</strong></td>
<td>0.7460</td>
<td><strong>0.6914</strong></td>
<td><strong>0.6982</strong></td>
</tr>
<tr>
<td>1:2</td>
<td>0.6466</td>
<td>0.7602</td>
<td>0.5058</td>
<td>0.5874</td>
</tr>
<tr>
<td>1:3</td>
<td>0.6675</td>
<td>0.7623</td>
<td>0.4562</td>
<td>0.5433</td>
</tr>
<tr>
<td>1:9</td>
<td>0.6175</td>
<td><strong>0.7753</strong></td>
<td>0.2139</td>
<td>0.3232</td>
</tr>
</tbody>
</table>

Rating Distribution Alignment

(a) Distribution on MovieLens

(b) Agent-simulated distribution
LLM-empowered Agents for Recommendation

Agent as User Simulator

- **Main idea**: using agents to simulate user behavior for real-world recommendation.
- RecAgent\(^1\), Agent4Rec\(^2\)

Agent for Recommendation

- **Main idea**: harnessing the powerful capabilities of LLMs, such as reasoning, reflection, planning and tool usage, for recommendation.
- RecMind\(^3\), InteRecAgent\(^4\), BiLLP\(^5\), Multi-Agent Collaboration\(^6\)

---


\(^5\) Wentao Shi et al. 2023. Large Language Models are Learnable Planners for Long-Term Recommendation. in SIGIR 2024.

\(^6\) Jiabao Fang et al. A Multi-Agent Conversational Recommender System. Arxiv 2024
Agent: RecMind

- **LLM-based agent for recommendation**
  - Traditional methods train and fine-tune models on task-specific datasets, struggle to leverage external knowledge and lack generalizability across tasks and domains.
  - Existing LLM4Rec methods primarily rely on internal knowledge in LLM weights.
  - RecMind fully utilizes strong planning and tool-using abilities of LLMs for recommendation.
Agent: RecMind

- **Planning ability**
  - To break complex tasks into smaller sub-tasks.
  - **Self-inspiring** to integrates multiple reasoning paths.

- **Tool-using ability**
  - **Database tool** to access domain-specific knowledge.
  - **Search tool** to access real-time information.
  - **Text summarization tool** to summarize lengthy texts.

- **Evaluation**
  - **Precision-oriented tasks** (rating prediction, direct recommendation, and sequential recommendation).
  - **Explainability-oriented tasks** (explanation generation and review summarization).

- **Result**
  RecMind can achieve performance comparable to the fully trained P5 model.

---


---

<table>
<thead>
<tr>
<th>Methods</th>
<th>Beauty</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HR@5</td>
<td>NDCG@5</td>
</tr>
<tr>
<td>S-Rec</td>
<td>0.0387</td>
<td>0.0244</td>
</tr>
<tr>
<td>SASRec</td>
<td>0.0401</td>
<td>0.0264</td>
</tr>
<tr>
<td>P5 (pre-trained expert, few-shot)</td>
<td>0.0495</td>
<td>0.0347</td>
</tr>
<tr>
<td>ChartGPT (zero-shot)</td>
<td>0.0689</td>
<td>0.0053</td>
</tr>
<tr>
<td>ChartGPT (few-shot)</td>
<td>0.0179</td>
<td>0.0124</td>
</tr>
<tr>
<td>RecMind-CasT (zero-shot)</td>
<td>0.0182</td>
<td>0.0139</td>
</tr>
<tr>
<td>RecMind-CasT (few-shot)</td>
<td>0.0349</td>
<td>0.0187</td>
</tr>
<tr>
<td>RecMind-ToT (BFS, zero-shot)</td>
<td>0.0297</td>
<td>0.0172</td>
</tr>
<tr>
<td>RecMind-ToT (BFS, few-shot)</td>
<td>0.0387</td>
<td>0.0235</td>
</tr>
<tr>
<td>RecMind-ToT (DFS, zero-shot)</td>
<td>0.0299</td>
<td>0.0168</td>
</tr>
<tr>
<td>RecMind-ToT (DFS, few-shot)</td>
<td>0.0365</td>
<td>0.0211</td>
</tr>
<tr>
<td>RecMind-SI (zero-shot)</td>
<td>0.0339</td>
<td>0.0200</td>
</tr>
<tr>
<td>RecMind-SI (few-shot)</td>
<td>0.0615</td>
<td>0.0289</td>
</tr>
</tbody>
</table>
Use Agent to simulate both user/items

Provide a collaborative reflection optimizing mechanism to optimize the user/item agents, and mutual update of user and item memory.

Junjie Zhang et al. 2023. Agentcf: Collaborative learning with autonomous language agents for recommender systems. in WWW 2024
Agent: AgentCF

Better performance and less influenced by bias than directly instructing LLM to rerank

Collaborative Reflection is effective to optimize the agent’s ability to distinguish positive/negative items

Junjie Zhang et al. 2023. Agentcf: Collaborative learning with autonomous language agents for recommender systems. in WWW 2024
Agent: BiLLLP

- Use LLM to make plans for long-term recommendations
- Utilize a bi-level learnable mechanism to learn macro-level guidance and micro-level personalized recommendation policies.
Agent: BiLLP

- Better long-term performance than traditional RL-based methods
- Better planning capabilities on long-tail items.
Different Agents Collaborate together for Conversational Recommendation

- The responder agent and planner agent collaboratively generate appropriate responses, while the reflection mechanism provides feedback and refined guidance to these agents.
Different agents can collaborate together for information delivery.

**Task:** Sequential Recommendation
User ID: 46
Candidate Item ID: [849, 376, 291, 125, 151, 132, 16, 814]
Rank the above candidates for the user.

**UserInfo** [UserID: 46]
**Finish:**
The user 46 is a 27-year-old female marketing professional who has interacted with a variety of movie genres including Drama, Sci-Fi, Adventure, Children, and Comedy.

**ItemInfo** [ItemID: 849]
**Finish:**
The movie 'Days of Thunder' (1990) has been interacted with by users of various ages and genders, indicating a broad appeal across different demographics.

**Search** [Phenomenon (film)] (ItemID: 125)
**LookUp** [Phenomenon (film), plot]
**Finish:**
Phenomenon is a 1996 American romantic fantasy drama film directed by Jon Turteltaub, written by Gerald Di Pego, in which George Malley, exhibits remarkable levels of intelligence and develops telekinesis after being hit by a ball of shining bright white lights...

**Last trial answer:** [151, 125, 814, 291, 376, 132, 849, 16]
**Correctness:** False
**Reason:**
The agent did not ensure that the response contained 8 ids, which was the same as the number of candidates in the question.

**Output:** [151, 125, 814, 291, 376, 132, 849, 16]
Collaborate Together can Solve Different Task

A demo for conversational recommendation via multi-agent collaboration

ChatSystem for Conversational Recommendation

I have watched the movie Schindler’s List recently. I am very touched by the movie. I wonder what other movies can teach me about history like this?

System is running...

- **Interpreter**: Summarize input...
  - Summarized text: user: I am very touched by the movie Schindler’s List.
- **Interpreter**: Finish with results:
  - Can you recommend other movies that can teach me about history like Schindler’s List?

Manager: Thought 1: There are many movies that can teach about history, but I need to consider the specific themes and time periods that the user is interested in.

Manager: Calling Searcher with movies about history...

- **Searcher**: Search for movies about history...
  - **Searcher**: Lookup history in document Skin: A History of Nudity in the Movies...
    - (Result 1 / 3) Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film. It was directed by Danny Wolf and executive produced by Jim "Mr. Skin" McBride, who is also interviewed in the film...
  - **Searcher**: Finish with results:
    - Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film.

Manager: Response from Searcher with movies about history:

- Search result: Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film.

Manager: Thought 2: Skin: A History of Nudity in the Movies is not relevant to the user’s request. I need to refine the search to find movies that specifically teach about historical events and themes.

Manager: Calling Searcher with historical movies similar to Schindler’s List...

- **Searcher**: Search for historical movies similar to Schindler’s List...
  - Found 3 documents. Their titles and summaries are (with the format title (summary)): Schindler’s List (Schindler’s List is a 1993 American epic historical drama film directed by Steven Spielberg and written by Steven...), Amistad (film) (Amistad is a 1997 American historical drama film directed by Steven Spielberg, based on the events in 1839 aboard the...), Steven Allan Spielberg (Steven Allan Spielberg (; born December 18, 1946) is an American film director, producer and screenwriter. A major figure of...)...
  - **Searcher**: Finish with results:
    - Schindler’s List, Amistad (film)

Manager: Response from Searcher with historical movies similar to Schindler’s List:

- Search result: Schindler’s List, Amistad (film)

Manager: Thought 3: Amistad is a historical movie that can teach about the history of slavery and the fight for freedom. I should recommend this movie to the user.

I recommend the movie Amistad, which teaches about the history of slavery and the fight for freedom.

---

Zhefan Wang et al. Multi-Agent Collaboration Framework for Recommender Systems. SIGIR 2024
Progress of LLM4Rec

Three dimensions:

- **Information modalities**: From the text modality to multiple modalities
- **Metrics**: From accuracy to trustworthiness such as privacy, fairness, etc.
- **How to utilize LLMs**: In-context learning, tuning, agent
Progress of LLM4Rec

- **Multimodal LLM4Rec**
  - GPT-4V for recommendation
  - Zero-shot recommendation ability
  - A series of qualitative test samples show remarkable performance
  - Reasons: robust visual-text comprehension capabilities and extensive general knowledge
  - Cons: lack response diversity.
  - **Future directions**: leverage multimodal user and item features for LLM-based recommendation
    - Tokenization with multimodal features
    - Model structure with with multimodal inputs and outputs
    - Optimization objectives to exclude multimodal noises
    - ...
Progress of LLM4Rec

Three dimensions:

- Information modalities
  - From the text modality to multiple modalities

- Metrics
  - From accuracy to trustworthiness such as privacy, fairness, etc.

- How to utilize LLMs
  - In-context learning
  - Tuning
  - Agent

- Fairness
- Robustness & OOD
- Privacy
- Safety
- Explainability

From the text modality to multiple modalities
Progress of LLM4Rec

Three dimensions:

Information modalities
From the text modality to multiple modalities

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.

How to utilize LLMs
In-context learning
Tuning
Agent

Fairness
Robustness & OOD
Privacy
Safety
Explainability

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.
User-side Fairness

- Does ChatGPT give fair recommendations to user with different sensitive attributes?

- We judge the fairness by comparing the similarity between the recommended results of different sensitive instructions and the neutral instructions.

- Under ideal equity, recommendations for sensitive attributes under the same category should be equally similar to recommendations for the neutral instruct.
User-side Fairness

- Dataset Construction.
  - Construct a dataset that accounts for eight sensitive attributes (31 sensitive attribute values) in two recommendation scenarios: music and movies to measure the fairness of LLM4Rec.

Template:

Netrual: “I am a fan of [names]. Please provide me with a list of K song/movie titles…”

Sensitive: “I am a/an [sensitive feature] fan of [names]. Please provide me with a list of K song/movie titles…”

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>middle aged, old, young</td>
</tr>
<tr>
<td></td>
<td>American, British, Brazilian</td>
</tr>
<tr>
<td>Country</td>
<td>Chinese, French, German, Japanese</td>
</tr>
<tr>
<td>Gender</td>
<td>boy, girl, male, female</td>
</tr>
<tr>
<td>Continent</td>
<td>African, Asian, American, doctor, student, teacher,</td>
</tr>
<tr>
<td>Occupation</td>
<td>worker, writer</td>
</tr>
<tr>
<td>Race</td>
<td>African American, black, white, yellow</td>
</tr>
<tr>
<td>Religion</td>
<td>Buddhist, Christian, Islamic</td>
</tr>
<tr>
<td>Physics</td>
<td>fat, thin</td>
</tr>
</tbody>
</table>

User-side Fairness

- Unfairness still exist in LLM4Rec

Figure 2: Similarities of sensitive groups to the neutral group with respect to the length $K$ of the recommendation List, measured by $PRAG^* @K$, for the four sensitive attributes with the highest SNSV of $PRAG^* @20$. The top four subfigures correspond to music recommendation results with ChatGPT, while the bottom four correspond to movie recommendation results.
User-side Fairness

- LLMs show implicit discrimination only according to user names

- **Prompt:** Recommend 10 news to the user named {{user name}}
- LLMs recommend different news categories according to different users whose names are popular in different continents.

RQ1: Why does implicit user unfairness exist?

- LLMs can infer sensitive attributes from user's non-sensitive attributes according to their wide world knowledge.

User-side Fairness

RQ2: How serious is implicit user unfairness?

Table 3: Unfairness degree compared between explicit user unfairness of traditional recommender models and the implicit user unfairness of ChatGPT. “Improv.” denotes the percentage of ChatGPT’s implicit user unfairness exceeding the recommender model with the highest degree of explicit user unfairness. Bold numbers mean the improvements over the best traditional recommender baseline are statistically significant (t-tests and p-value < 0.05).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U-NDCG@1</td>
<td>0.17</td>
<td>0.225</td>
<td>0.025</td>
<td></td>
<td>0.305</td>
<td>35.6%</td>
<td>0.16</td>
<td>0.045</td>
<td></td>
<td>0.365</td>
<td>46.0%</td>
</tr>
<tr>
<td>U-NDCG@3</td>
<td>0.177</td>
<td>0.183</td>
<td>0.024</td>
<td></td>
<td>0.363</td>
<td>98.4%</td>
<td>0.115</td>
<td>0.041</td>
<td></td>
<td>0.366</td>
<td>70.2%</td>
</tr>
<tr>
<td>U-NDCG@5</td>
<td>0.104</td>
<td>0.12</td>
<td>0.016</td>
<td></td>
<td>0.203</td>
<td>69.2%</td>
<td>0.08</td>
<td>0.025</td>
<td></td>
<td>0.215</td>
<td>60.6%</td>
</tr>
<tr>
<td>U-MRR@1</td>
<td>0.17</td>
<td>0.225</td>
<td>0.025</td>
<td></td>
<td>0.305</td>
<td>35.6%</td>
<td>0.16</td>
<td>0.045</td>
<td></td>
<td>0.365</td>
<td>46.0%</td>
</tr>
<tr>
<td>U-MRR@3</td>
<td>0.173</td>
<td>0.193</td>
<td>0.026</td>
<td></td>
<td>0.348</td>
<td>80.3%</td>
<td>0.126</td>
<td>0.042</td>
<td></td>
<td>0.368</td>
<td>64.3%</td>
</tr>
<tr>
<td>U-MRR@5</td>
<td>0.136</td>
<td>0.158</td>
<td>0.021</td>
<td></td>
<td>0.264</td>
<td>67.1%</td>
<td>0.106</td>
<td>0.033</td>
<td></td>
<td>0.288</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

• More serious than traditional recommender models!

RQ3: What are the long-term impacts?

• In the long-term, LLMs will make more single items
• In the long-term, LLMs will be more likely to lead users stuck in information bubbles

Item-side Fairness

- **Item-side fairness**
  - LLM-based recommendation systems exhibit *unique characteristics (like recommend based on semantic)* compared to conventional recommendation systems.
  - Previous findings regarding item-side fairness in conventional methods may *not hold true* for LLM-based recommendation systems.
  - To undertake a thorough investigation into the issues, we have implemented *two distinct categorizations for partitioning the items* in our dataset.

Item-side Fairness

- Item-side fairness (Popularity)
  - The results indicate LLM-based recommender system excessively recommended group with the highest level of popularity.
  - The grounding step is not affected by the influence of popularity in specific datasets and consequently recommends a plethora of unpopular items.

**Item-side Fairness**

- **Item-side fairness (Genre)**
  - The high-popularity genre groups would be over-recommended (Pos GU), while low-popularity genres tend to be overlooked (Neg GU).

- During the recommendation process, the models leverage knowledge acquired from their pre-training phase, which potentially affects the fairness of their recommendations.

---

Progress of LLM4Rec

Three dimensions:

- How to utilize LLMs
  - In-context learning
  - Tuning
  - Agent

- Information modalities
  - From the text modality to multiple modalities

- Metrics
  - From accuracy to trustworthiness such as privacy, fairness, etc.

- Fairness
- Robustness & OOD
- Privacy
- Safety
- Explainability

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.
LLM4Rec is robust to unintentionally generated typos.

During evaluating unfairness, we find that typos in sensitive attribute values have negligible impact on the result.

Robustness & OOD

- Out-of-distribution (OOD) generalization

- Learning from movie scenario can directly recommend on books, and vice versa making the LLMRec has strong OOD generalization ability.

Keqin Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023
Progress of LLM4Rec

Three dimensions:

Information modalities
From the text modality to multiple modalities

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.

How to utilize LLMs
In-context learning
Tuning
Agent

Fairness
Robustness & OOD
Privacy
Safety
Explainability

From the text modality to multiple modalities
Privacy Unlearning

- Challenges for LLMRec Unlearning
  - Needs exact unlearning to protect user privacy
  - Reasonable inference time enables timely responses to user demands

- Existing works for LLM Unlearning
  - Gradient update
  - In-context Unlearning
  - Simulates data labels

  - ALL those methods can't handle challenge 1.

- Data-partition base retraining paradigm
  - Devide data into multi-groups
  - Train each sub-model
  - Aggregate the output of each sub-model

  - This paradigm can't handle challenge 2.

Zhiyu Hu et al. Exact and Efficient Unlearning for Large Language Model-based Recommendation. Arxiv 2024
• Partition data based on semantics
• Differing from the previous paradigm, we leverage adapter weight aggregation during the inference phase.

Zhiyu Hu et al. Exact and Efficient Unlearning for Large Language Model-based Recommendation. Arxiv 2024
APA exhibits less performance loss compared to the reference Retraining method and can even bring improvements.

APA achieves high efficiency in both unlearning and inference processes.
E2URec

- E2URec aim to achieve unlearning by using two teachers.
- Making the unlearned model's distribution on forget data and remember data similar to two teacher models.

**Forgetting Teacher**

- Using Augmented Model trained on forgotten data to estimate the forgetting teacher

**Unlearning with Teachers**

- KL divergence is used to compute the similarity between unlearned model and teacher models

\[
\min_{\theta} \text{KL}\left(\mathcal{M}_f(\mathcal{D}_f) \parallel \mathcal{M}_u(\mathcal{D}_f; \theta)\right)
\]

\[
\min_{\theta} \text{KL}\left(\mathcal{M}_r(\mathcal{D}_r) \parallel \mathcal{M}_u(\mathcal{D}_r; \theta)\right)
\]

Federated Learning

- **Motivation of Incorporating Federated Learning**
  - Preserve data privacy when fine-tuning LLMs with user behavior data

- **Challenge of Incorporating Federated Learning**
  - Exacerbated Client Performance Imbalance
  - Substantial Client Resource Cost

Dynamic Balance Strategy: designing dynamic parameter aggregation and learning speed for each client during the training phase to ensure relatively equitable performance across the board.

Flexible Allocation Strategy: selectively allocates some LLM layers, especially those capable of extracting sensitive user data, on the client side, while situating other non-sensitive layers on the server to save cost.

Progress of LLM4Rec

Three dimensions:

- Information modalities: From the text modality to multiple modalities
- Metrics: From accuracy to trustworthiness such as privacy, fairness, etc.
- How to utilize LLMs: In-context learning, Tuning, Agent

Fairness, Robustness & OOD, Privacy, Safety, Explainability
Text-centric paradigm raises new security issue of RS:
Attackers can significantly boost an item's exposure by merely altering its textual content.

- From text perspective
- Not involve training
- Hard to be detected

Safety

Attack:
Use GPT/textual attack methodologies to rewrite item description until reach the goal.

Prompt 1: You are a marketing expert that helps to promote the product selling. Rewrite the product title in <MaxLen> words to keep its body the same but more attractive to customers: <ItemTitle>.

Potential Defend:

Re-writing Prompt: Correct possible grammar, spelling and word substitution errors in the product title (directly output the revised title only): <AdversarialTitle>

<table>
<thead>
<tr>
<th>Model</th>
<th>Text</th>
<th>Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>Fisher-Price Fun-2-Learn Smart Tablet</td>
<td>0.0076</td>
</tr>
<tr>
<td>Trivial GPT</td>
<td>Fisher-Price Fun-2-Learn Smart Tablet</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>Interactive Learning Tablet for Kids</td>
<td>0.0335</td>
</tr>
<tr>
<td>DeepwordBug</td>
<td>Fisher-Price Fun-2-Learn Smar Ttable</td>
<td>0.0335</td>
</tr>
<tr>
<td>+Defense</td>
<td>Fisher-Price Fun-2-Learn Smart Tablet</td>
<td>0.0076</td>
</tr>
<tr>
<td>PunAttack</td>
<td>Fisher-Price Fun-2-Learn Sm’art Tablet</td>
<td>0.0285</td>
</tr>
<tr>
<td>+Defense</td>
<td>Fisher-Price Fun-2-Learn Smart Tablet</td>
<td>0.0076</td>
</tr>
<tr>
<td>Textfooler</td>
<td>Fisher-Price Fun-2-Learn Canny Table</td>
<td>0.0768</td>
</tr>
<tr>
<td>+Defense</td>
<td>Fisher-Price Fun-2-Learn Canine Table</td>
<td>0.0756</td>
</tr>
<tr>
<td>BertAttack</td>
<td>Fisher-Price Fun-2-Learn this Tablet</td>
<td>0.0262</td>
</tr>
<tr>
<td>+Defense</td>
<td>Fisher-Price Fun-2-Learn Tablet</td>
<td>0.0190</td>
</tr>
</tbody>
</table>

Progress of LLM4Rec

Three dimensions:

Metrics
From accuracy to trustworthiness such as privacy, fairness, etc.

Information modalities
From the text modality to multiple modalities

How to utilize LLMs
In-context learning
Tuning
Agent

Fairness
Robustness & OOD
Privacy
Safety
Explainability

From the text modality to multiple modalities
**Explainability**

- Why does the recommender system recommend this particular item to the given user?
- LLM could directly generate explanations for their recommendations:

  Q1: Hello! How can I help you today?
  A1: Hello! How can I help you today?
  Q2: I want to watch a movie now, do you have any recommendations?
  A2: Based on your preference for action, adventure, comedy, and romance movies, I would recommend the following top 5 movies:
  1. Star Wars (1977) - An epic space opera that...
  2. Star Trek: First Contact (1996) - A science-fiction adventure film that...
  3. Fargo (1996) - A dark comedy-crime film following...
  4. The Fifth Element (1997) - A visually stunning science-fiction film about...
  5. Face/Off (1997) - An action-thriller film where...

  Q3: I've heard of Fargo (1996). It looks very interesting. Why would you recommend this movie to me? What do you think would appeal to me about this movie?
  A3: I'm glad you're interested in Fargo (1996)! Based on the historical records, it seems that you appreciate movies with a mix of genres, including drama, thriller, and comedy. Fargo is a perfect example of a film that blends these elements effectively. Here are a few reasons why I believe it would appeal to you:
  1. Unique storytelling: Fargo is a dark comedy-crime film that...
  2. Memorable characters: The film introduces a memorable cast of characters, including...
  3. Dark humor: Fargo is known for its dark humor and satirical take on crime...
  4. Critical acclaim: Fargo received widespread acclaim from critics and audiences alike, winning two Academy Awards...
  5. The Coen Brothers: Fargo is directed by the talented Coen Brothers, ...


113
Finetune LLM for Rec Explanation

- Design different tasks to finetune LLM for Recommendation Explanation
- Besides finetuning for recommendation performance, RecExplainer finetunes LLM on different task related to recommendation explanation, such as Item discrimination and history reconstruction.

![Diagram showing the process of finetuning LLM for recommendation explanation.](image-url)
Outline

• Introduction
• Background: LM & LM4Rec
• Development of LLMs
• Progress of LLM4Rec

• **Open Problems**
  • Modeling
  • Cost
  • Evaluation

• Future Direction & Conclusions
Open Problems & Challenges

Three aspects:

Modeling

LLM: modeling text/language

RecSys: modeling behaviors

Cost

LLM: high cost/delay

LLM: Trained on many data, text-focused, language

RecSys: cost sensitive

Evaluation

LLM: 

RecSys research: interactions, offline, anonymous data

Evaluation?
Modeling: User/Item Representation

- Recommendation: user behavior modeling
- How should we represent user behaviors (represent users/items) in LLM4Rec?

**LLM4Rec methods**

User/Item: Text

- LLM
- Generated Response
- Answer Parsing

A user... movies titled ‘xxx’... the movie titled ‘xxx’?

**Traditional methods**

User/Item: features + ID

- Textually similar item may have distinct collab. info.
- May lack of some information

LLMs are constructed using texts, making the representation of users/items in texts the natural choice.

Features (content) alone are insufficient to depict users and items, mainly behavioral similarities (collaborative info). IDs are utilized.
Modeling: User/Item Representation

Integrate collaborative information:
• Why?

LLM Rec vs Traditional CF Model:
#:Excellent at old-start scenarios  #: Poor at warm-start scenarios

Modeling: User/Item Representation

Integrate collaborative information: How?

**#1: learn user/item embedding by fitting interactions**

- **LLM**
- Add tokens to represent users and items in LLM
- Learn LLM token embeddings by fitting interaction data
- Large space, low learning efficacy
- Design better tokenization

**#2: Feed the collaborative information extracted by external models into LLM**

- Extract collaborative information with traditional models
- Feed the extracted information into LLMs
- Focus on how to feed the info.
Modeling: User/Item Representation

Integrate collaborative information: feed external collaborative information into LLM

- Work#1: CoLLM —— mapping collaborative embeddings into LLM’s Latent space

  ![Diagram of CoLLM integration](image)

  Latent space!!

  - **Prompt construction**: add <UserID> and <TargetID> for placing the Collab. Info.
  - **Hybrid Encoding**:
    - text: tokenization & LLM emb Lookup;
    - user/item ID: CIE --- extract info with collab. model (low rank), then map it to the token embedding space
  - LLM prediction: add a LoRA module for recommendation task learning

Integrate collaborative information: feed external collaborative information into LLM

- **Work#1: CoLLM** —— mapping collaborative embeddings into LLM’s Latent space

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ML-1M</th>
<th>Amazon-Book</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>UAUC</td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collab.</td>
<td>MF</td>
<td>0.6482</td>
</tr>
<tr>
<td></td>
<td>LightGCN</td>
<td>0.5959</td>
</tr>
<tr>
<td></td>
<td>SASRec</td>
<td>0.7078</td>
</tr>
<tr>
<td>LLMRec</td>
<td>ICL</td>
<td>0.5320</td>
</tr>
<tr>
<td></td>
<td>Soft-Prompt</td>
<td>0.7071</td>
</tr>
<tr>
<td></td>
<td>TALLRec</td>
<td>0.7097</td>
</tr>
<tr>
<td>Ours</td>
<td>CoLLM-MF</td>
<td>0.7295</td>
</tr>
<tr>
<td></td>
<td>CoLLM-LightGCN</td>
<td>0.7100</td>
</tr>
<tr>
<td></td>
<td>CoLLM-SASRec</td>
<td>0.7235</td>
</tr>
</tbody>
</table>

- CoLLM brings performance improvements over traditional models and current LLM Rec in most cases
- CoLLM significantly improves the warm performance of LLM4Rec, while ensuring cold performance

Integrate collaborative information: feed external collaborative information into LLM

- Work#2: BinLLM —— Encoding collaborative embeddings in a text-like format for LLM
  - Transform the collaborative embeddings into binarized sequence, treating them as textual features directly usable by LLMs
  - LLMs could naturally perform bitwise operations
  - Binarizing collaborative embeddings could keep performance.

Feed collaborative information into prompts
Modeling: User/Item Representation

Integrate collaborative information: feed external collaborative information into LLM

• More works

Modeling: User/Item Representation

Integrate collaborative information: learn user/item-specific token embedding

- learn user/item embedding by fitting interactions
- Add new tokens to represent users and items in LLM
- Learn LLM token embeddings by fitting interaction data
- Default choice: Random IDs as tokens
- Issues:
  - Large learning space --- low learning efficacy
  - Semantic gaps between text tokens and recommendation tokens
  - Generalization issues --- cannot deal with new items
Integrate collaborative information: learn user/item-specific token embedding

- Work#1: Collaborative indexing: Clustering collaborative information to create IDs
  - Generate collaborative embeddings
  - Hierarchically cluster the collaborative embedding
  - generate IDs based on category indices

Advantages:
1) Add constraints on item IDs
2) Reduce the token spaces
3) Increase the learning efficacy.

A set of sub-IDs

Modeling: User/Item Representation

Integrate collaborative information: **learn user/item-specific token embedding**

- **Work#2**: Semantic-aware ID (Tiger/LC-Rec): quantizing text embedding to generate IDs

  - Convert text content information into embeddings
  - Quantization: represent the text embedding with several sub-embeddings, generating semantic ID
  - Several sub-IDs form a semantic ID

**Advantages:**
1) Reduce the token spaces, $N \rightarrow K \cdot N^{1/K}$
2) Could deal with new items

---

Integrate collaborative information: learn user/item-specific token embedding

- **Summary of tokenizer (item-side):**

1. Textual Identifiers
   - BigRec
   - GenRec
   - Lack of collab. info.
   - Could deal with new items

2. Vanilla ID
   - Random ID
   - Large space, hard to learn
   - Good at collab. info.
   - Cannot deal with new items

3. Collaborative ID
   - CID
   - Small space
   - Good at collab. info.
   - Cannot deal with new items

4. Semantic ID
   - TIGER
   - LC-Rec
   - Small space
   - Limited collaborative info.
   - Could deal with new items

- **Open Problem:**
  - Tokenization on user behaviors
  - Tokenization on cross-domain items

Modeling: Lifelong Modeling

• Users are anticipated to engage with the recommender system continuously
• Raise the need of lifelong behavior modeling for users

Lifelong sequential behavior modeling
• The length of historical interaction sequences grows significantly, easily exceeding 1000
• How to model such long sequence effectively?

Continual learning
• User interests drift with time going
• How to continuously/incremental learn user interests?
Lifelong sequential behavior modeling:

- A longer history signifies **richer personalization information**, and modeling this can lead to heightened prediction accuracy.

An example in the advertising system in Alibaba.

Lifelong sequential behavior modeling:

LLM cannot effectively model long user Behavior sequence

- Extending user behavior sequences doesn't necessarily enhance recommendation performance, even if the input length is far below the length limit of LLMs (e.g., Vicuna-13B has an upper limit of 2048 tokens).

Li et al. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. WWW 2024.
Lifelong sequential behavior modeling:

- **Work#1: ReLLa** --- just retrieve most (semantically) similar items from the history
  
  **Step 1:** For a **target item**, retrieve the top-K **semantically similar items from the history**, forming a new sample
  
  **Step 2:** Leverage the original sample and new sample to fine tune LLM for recommendation

- Limitations: heavily depends on “target attention, not applicable when the input lacks target items.
- Future: may need to explore other solutions like memory.

Li et al. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. WWW, 2024.
Modeling: Lifelong Modeling

Continual learning:

- How to incrementally learn user interests?
- There is work [1] studying the common used methods: periodic retraining

Periodically update TALLRec does not bring significant performance improvements.

LLM4Rec may struggle to capture short-term preferences in the latest data with traditional periodic updates, limiting performance improvement.

Outline

• Introduction
• Background: LM & LM4Rec
• Development of LLMs
• Progress of LLM4Rec
• Open Problems
  • Modeling
  • Cost
  • Evaluation
• Future Direction & Conclusions
Cost

- The income-cost trade-off is sensitive for recommendation
- Deployment cost of LLM4Rec is high

LLM Parameters: tens/hundreds of billions

Training and inference:
- High demand on GPUs/Memory
- Slow

How to reduce the cost?
Cost: Training

One exploration: Data-efficient training

- Fine-tuning LLM is necessary
  - LLMs are not particularly trained on recommendation data
- LLM fine-tuning is expensive, e.g., high computational costs, time-consuming
- Few-shot fine-tuning is a promising solution
- Data pruning for efficient LLM-based recommendation
  - identify representative samples tailored for LLMs

One exploration: Data-efficient training

- Two objectives for data pruning
  - high accuracy: select the samples that can lead to higher performance -> influence score
  - high efficiency: emphasize the low costs of the data pruning process
    - surrogate model to improve efficiency
    - effort score to bridge between surrogate model and LLMs

- Experimental results
  - fine-tune with 1024 samples

<table>
<thead>
<tr>
<th></th>
<th>R@10↑</th>
<th>R@20↑</th>
<th>Games (N@10)↑</th>
<th>Games (N@20)↑</th>
<th>Time↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>0.0169</td>
<td>0.0233</td>
<td>0.0102</td>
<td>0.0120</td>
<td>36.87h</td>
</tr>
<tr>
<td>DEALRec</td>
<td>0.0181</td>
<td>0.0276</td>
<td>0.0115</td>
<td>0.0142</td>
<td>1.67h</td>
</tr>
<tr>
<td>% Improve.</td>
<td>7.10%</td>
<td>18.45%</td>
<td>12.75%</td>
<td>18.33%</td>
<td>-95.47%</td>
</tr>
</tbody>
</table>

- Increasing samples from 0.2% to 4% of all training data

Lin et al. Data-efficient fine-tuning for LLM-based recommendation. SIGIR'24.
Cost: Inference

One solution: distillation

Distill LLM’s knowledge to smaller models and utilize small models for inference

- Work#1: distill recommendation results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>HR@20</th>
<th>NDCG@20</th>
<th>Inference time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Games</td>
<td>DROS</td>
<td>0.0473</td>
<td>0.0267</td>
<td>1.8s</td>
</tr>
<tr>
<td></td>
<td>BIGRec</td>
<td>0.0532</td>
<td>0.0341</td>
<td>2.3×10^4s</td>
</tr>
<tr>
<td></td>
<td>Gain</td>
<td>+12.47%</td>
<td>+27.72%</td>
<td>-1.3 × 10^6%</td>
</tr>
<tr>
<td>Toys</td>
<td>DROS</td>
<td>0.0231</td>
<td>0.0144</td>
<td>1.6s</td>
</tr>
<tr>
<td></td>
<td>BIGRec</td>
<td>0.0420</td>
<td>0.0207</td>
<td>1.1×10^4s</td>
</tr>
<tr>
<td></td>
<td>Gain</td>
<td>+81.82%</td>
<td>+43.75%</td>
<td>-6.8 × 10^5%</td>
</tr>
</tbody>
</table>

The inference latency of BIGRec far exceeds that of DROS.

- **Distillation challenges:**
  - 1) The teacher’s knowledge may not always be reliable.
  - 2) The divergence in semantic space poses a challenge to distill the knowledge from embeddings.

Cost: Inference

One solution: distillation
Distill LLM’s knowledge to smaller models and utilize small models for inference

- Work#1: distill recommendation results

  - Importance-aware Ranking Distillation
  filter reliable and student-friendly knowledge by weighting instances

  - Confidence of LLMs
  The distance between the generated descriptions with the target item
  Teacher-Student Consensus
  The items recommended by both teacher and student are more likely to be positive

  - Ranking Position
  Higher ranked items by teachers are more reliable

Collaborative Embedding Distillation
integrate knowledge from teacher embeddings with student’s

Cost: Inference

One solution: distillation
Distill LLM’s knowledge to smaller models and utilize small models for inference
• Work#2: distill recommendation rationales

- Distill recommendation rationale from ChatGPT to Llama-7B
- Empowering recommendation with rationale embedding
- Combining the rationale embedding and item description embedding for prediction

Wang et al., Can Small Language Models be Good Reasoners for Sequential Recommendation? ArXiv 2024.
Outline

- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- **Open Problems**
  - Modeling
  - Cost
  - Evaluation
- Future Direction & Conclusions
Evaluation: Data Issues

- **Challenge #1: Lack of data for evaluation**
  - Most of benchmarks are proposed ahead of pre-training stage of LLMs, e.g., ChatGPT, LLaMA.
  
  - The information of recommendation datasets (e.g., reviews,) may be included in LLMs.
  
  - Existing works usually did not discuss this.
  
  - Evaluations on the data that is not included in pretraining data of LLMs.
Evaluation: Data Issues

- **Challenge#1: Lack of data for evaluation**
  - **Insufficient features**
    - Lack of raw feature
    - Anonymous (e.g., just feature ID)
    - Lack of content (e.g., video content)
    - Currently, many works just utilize titles
  - Underutilization of LLM capabilities;
  - Underassessment of the effectiveness of LLM4Rec

- **Data homogeneity**
  - content homogeneity: mostly from E-commerce platform / entertaining content or places
  - biased user distributions: mostly from China and U.S.
  - Not comprehensive evaluation
  - Biased evaluation
Challenge#2: Evaluate interactive recommendation

Conversational recommendation

- provide personalized recommendation via multi-turn dialogs in natural language
- focus on conversational quality and recommendation quality

Issues of traditional evaluation:

- Simulated users are overly simplified representations of human users
- Conversations are often vague about the user preference, but not focus on exactly match the ground-truth items
- Evaluation protocol is based on fixed conversations, but the conversation could be diverging.

New evaluation: simulation with LLM-based agents?

- Challenges: how to design simulators is still an open problem.
Evaluation: Interactive RecSys

- Challenge#2: Evaluate interactive recommendation

  - Long-term recommendation
    - Multi-turn user-system interactions
    - Focus on long-term user engagement, e.g., user retention

  - How to evaluate long-term engagement is a big challenge.
    - We have not feedback about the unseen interaction trajectory
    - Evaluation with agent-based simulator is a potential solution
Outline

• Introduction
• Background: LM & LM4Rec
• Development of LLMs
• Progress of LLM4Rec
• Open Problems
• Future Direction & Conclusions
Progress of LLM4Rec

Three dimensions:

- **Metrics**
  - From accuracy to **trustworthiness**
  - such as privacy, fairness, etc.

- **Information modalities**
  - From the text modality to **multiple modalities**

- **How to utilize LLMs**
  - In-context learning
  - Tuning
  - Agent
Open Problems

Three aspects:

**Modeling**
LLM: modeling text/language
RecSys: modeling behaviors

**Cost**
LLM: high cost/delay
- computation/
- memory-
- costly
- Real-time,
- focus on
- cost
RecSys: cost sensitive

**Evaluation**
LLM: Trained on many data,
- text-focused, language
RecSys research: interactions,
- offline, anonymous data

Gap
Gap
Evaluation?
Generative Recommendation Paradigm

Generative AI for recommendation

- Personalized **content generation**, including item repurposing and creation.
  - Application: News, fashion products, micro-videos, virtual products in games, etc.

Instructor:
- Pre-process user instructions and feedback to guide the content generation of the AI generator.

AI Editor:
- Refine or repurpose existing items according to personalized user instructions and feedback.
- External facts and knowledge might be used for content generation.

AI Creator:
- Generate new items based on personalized user instructions and feedback.

AI Checker:
- Generation quality checks.
- Trustworthiness checks.

Applicable to many domains, including images, micro-videos, movies, news, books, and even products (for manufacture).

Figure 4: A demonstration of GeneRec. The instructor collects user instructions and feedback to guide content generation. The AI editor aims to repurpose existing items in the item corpus while the AI creator directly creates new items.

Wenjie Wang et al. Generative Recommendation: Towards Next-generation Recommender Paradigm. arxiv 2023
Generative Recommendation in Fashion Domain

The Evolution of Fashion Outfit Recommendation

Generative Outfit Recommendation

Objective: generating a set of new personalized fashion products to compose a visually compatible outfit catering to users’ fashion tastes.

Practical Implementation: retrieve or customize
Generative Recommendation Paradigm

- **Experiments**
  - **Datasets:** iFashion, Polyvore-U
  - **Baselines:** generative models, retrieval-based models
  - **Tasks:** personalized Fill-In-The-Blank (PFITB), GOR
  - **Evaluation**
    - **Quantitative Evaluation**
    - **Human-involved Qualitative Evaluation**
      - on Amazon Mechanical Turk

Table 5: The human-involved qualitative evaluation results, where “±” denotes 95% confidence interval. DiFashion is consistently preferred (≥ 50%) over the baselines across all evaluation metrics for both PFITB and GOR tasks.

<table>
<thead>
<tr>
<th></th>
<th>DiFashion</th>
<th>Fidelity</th>
<th>Compatibility</th>
<th>Personalization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PFITB</strong></td>
<td>SD-v1.5</td>
<td>64.08±3.08%</td>
<td>60.44±2.42%</td>
<td>68.32±3.47%</td>
</tr>
<tr>
<td></td>
<td>SD-v2</td>
<td>70.04±4.16%</td>
<td>57.48±1.90%</td>
<td>66.40±3.39%</td>
</tr>
<tr>
<td><strong>GOR</strong></td>
<td>SD-v1.5</td>
<td>61.56±1.93%</td>
<td>61.20±2.00%</td>
<td>60.80±2.57%</td>
</tr>
<tr>
<td></td>
<td>SD-v2</td>
<td>66.52±2.15%</td>
<td>60.56±1.88%</td>
<td>63.72±1.95%</td>
</tr>
</tbody>
</table>

Existing agent platforms such as GPTs (OpenAI), Poe (Quora), and DouBao (ByteDance) possess a vast number of LLM-based agents.

How to recommend LLM-based Agent to the user?

Different from Items in Traditional Recommender System, LLM-based Agent holds the potential to extend the format of information carriers and the way of information exchange.

- Formulate new Information System
- New Rec paradigm Rec4Agentverse
In Rec4Agentverse, the relationship between user, Agent Item and Agent Recommender may be much closer. Agent Recommender can collaborate with Agent Items to affect the information flow of users and offer personalized information services.
Three stages of Rec4Agentverse. The bidirectional arrows depicted in the Figure symbolize the flow of information.

- **User-Agent interaction stage:** Information flows between the user and Agent Item.
- **Agent-Recommender Collaboration stage:** Information flows between Agent Item and Agent Recommender.
- **Agents Collaboration stage:** Information flows between Agent Items.
The scaling properties of the CTR model have been verified, showing excellent performance on both internal and open-source data.

Demonstrates the possibility of increasing the size of CTR models through clever structural design and appropriate scaling settings.

 Exhibits better scaling performance than previous models.

Buyun Zhang et al., 2024 Wukong: Towards a Scaling Law for Large-Scale Recommendation arxiv 2024
Generative Recommender

New model architecture and feature processing methods.

Jiaqi Zhai et al., 2024 Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations ICML 2024
Better performance than traditional models on in-house data and open source data (Above two table)

Far more scaling ability than traditional DLRMs

Jiaqi Zhai et al., 2024 Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations ICML 2024
Large Behaviour Model

- **What we have know?**
  - Scaling Law remains effective on industrial-scale data when combined with an appropriate architecture in the context of recommendation scenario.
  - When the model is large enough and captures high-order information, it exhibits a certain generalization ability across scenes and domains.
  - Generative recommender is more stronger than traditional methods.

- **What we haven’t know?**
  - How to integrate world knowledge, and whether it can be combined with LLMs.
  - In addition to features and simple action, how do we model more complex short-term and long-term user behaviors? And how is the scalability of these behaviors manifested?
  - How to model the shared information between items/users and items/users?
• Social media AI (RecSys) already embed values --- maximize each user's individual experience---as predicted through likes

• It can harm societal values --- wellbeing, social capital, mitigating harm to minoritized groups, democracy, and maintaining pro-social norms.

• Could we directly embed societal values into RecSys?

Social sciences craft rigorous definitions & measurement of values  

Engineering translates the definitions into replicable AI models  

Field experiments study the behavioral effects of the AI models
Thanks for Your Listening!

**Tutorial** on Large Language Models for Recommendation: Progresses and Future Direction

Find our slides at
https://generative-rec.github.io/tutorial/

**Survey:** A Survey of Generative Search and Recommendation in the Era of Large Language Models
https://arxiv.org/pdf/2404.16924

Follow our WeChat account “智荐阁”!
The immense ability of LLMs may exceed the capabilities of traditional recommendation benchmark!

- The LLM may recommend items that are not in the dataset but are in line with user's real preference, how will it be evaluated?

- The LLM may recommend non-existent but meaningful items that meet the user's preferences. How will this situation be evaluated?
The immense ability of LLMs may exceed the capabilities of traditional recommendation benchmark!