

# Tutorial on Large Language Models for Recommendation



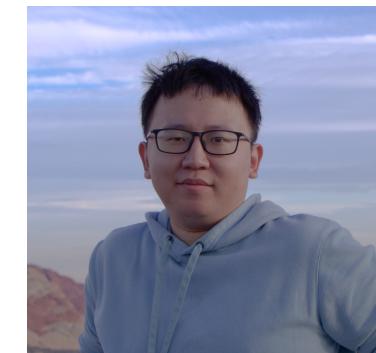
Wenyue Hua

Rutgers University  
[wenyue.hua@rutgers.edu](mailto:wenyue.hua@rutgers.edu)



Lei Li

Hong Kong Baptist University  
[csleili@comp.hkbu.edu.hk](mailto:csleili@comp.hkbu.edu.hk)



Shuyuan Xu

Rutgers University  
[shuyuan.xu@rutgers.edu](mailto:shuyuan.xu@rutgers.edu)



Li Chen

Hong Kong Baptist University  
[lichen@comp.hkbu.edu.hk](mailto:lichen@comp.hkbu.edu.hk)



Yongfeng Zhang

Rutgers University  
[yongfeng.zhang@rutgers.edu](mailto:yongfeng.zhang@rutgers.edu)

## Outline

- Background and Introduction
- Large Language Models for Recommendation
- Trustworthy LLMs for Recommendation
- Hands-on Demo of LLM-RecSys Development based on OpenP5
- Summary



# Recommender Systems are Everywhere

- Influence our daily life by providing personalized services

E-commerce



Social Networks



News Feeding



Search Engine



Navigation



Travel Planning



Professional Networks



Healthcare

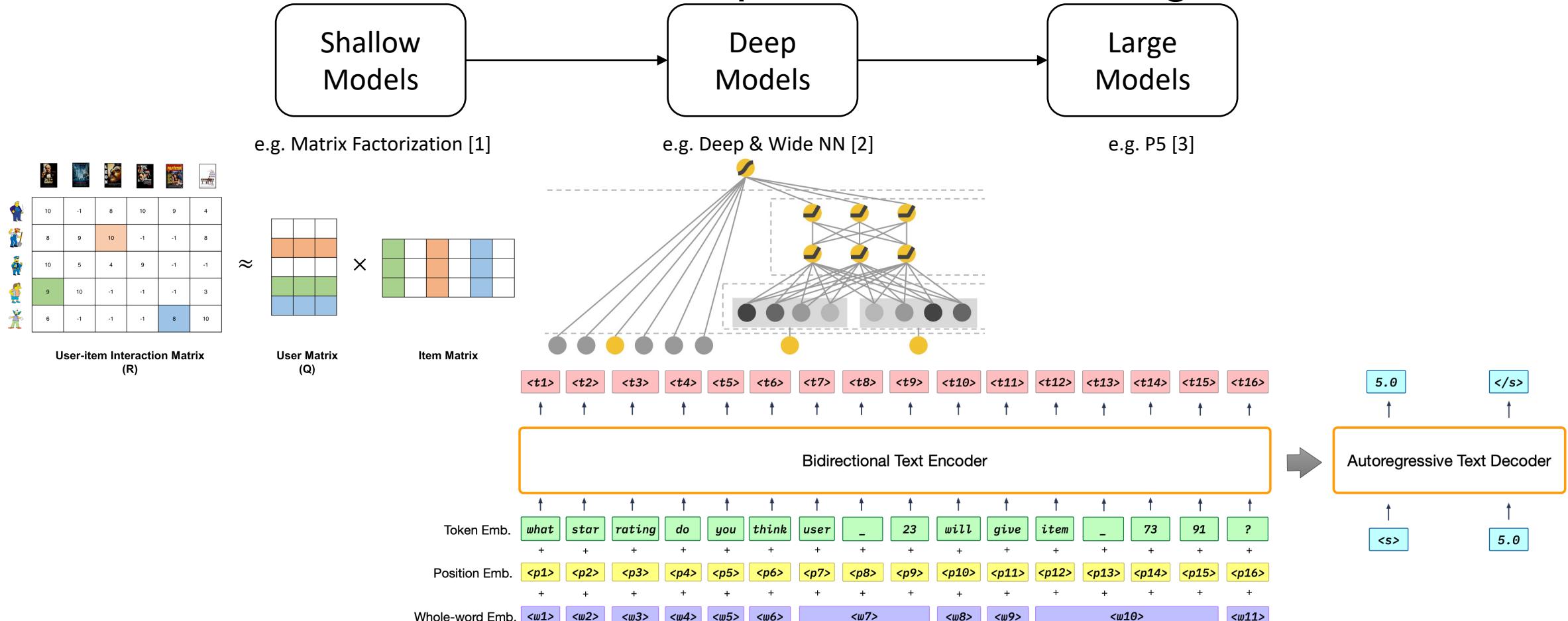


Online Education



# Technical Advancement of Recommender Systems

- From Shallow Model, to Deep Model, and to Large Model



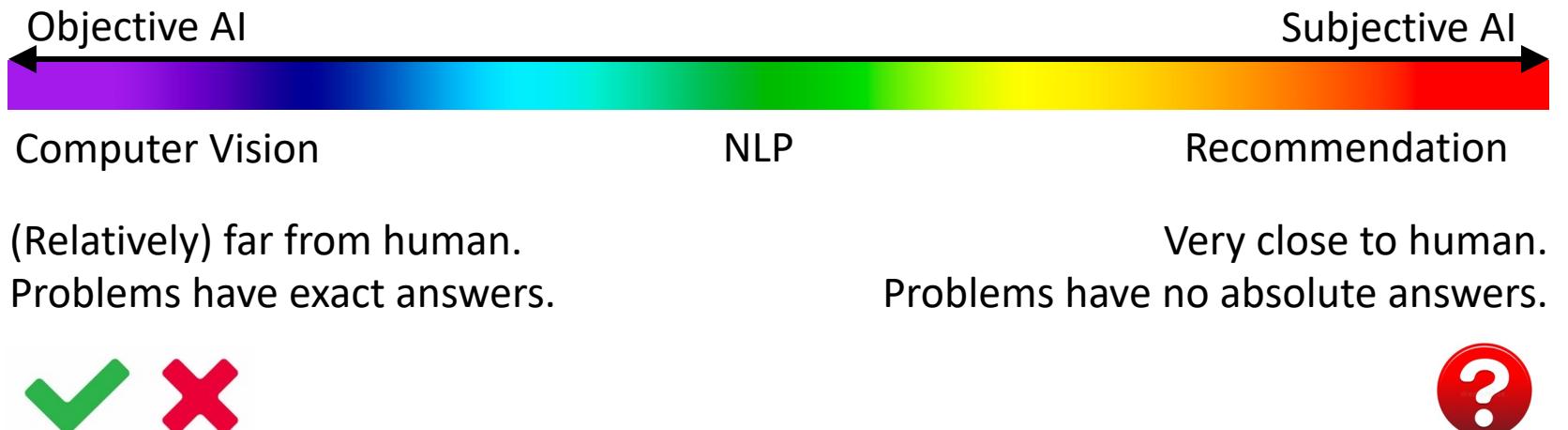
[1] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42, no. 8 (2009): 30-37.

[2] Cheng, Heng-Tze, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson et al. "Wide & deep learning for recommender systems." DLRS 2016.

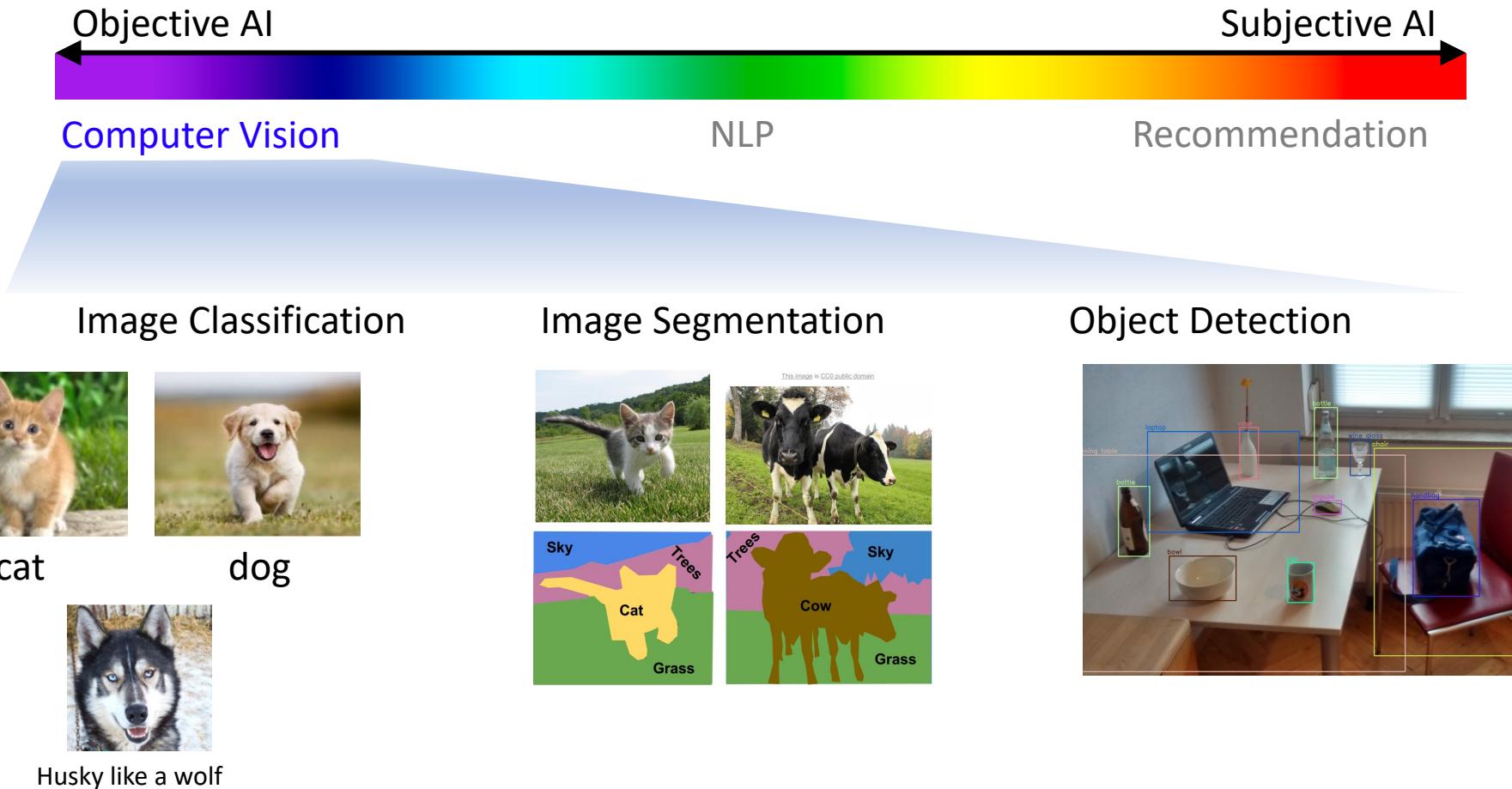
[3] Geng, Shijie, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)." RecSys 2022.

# Objective AI vs. Subjective AI

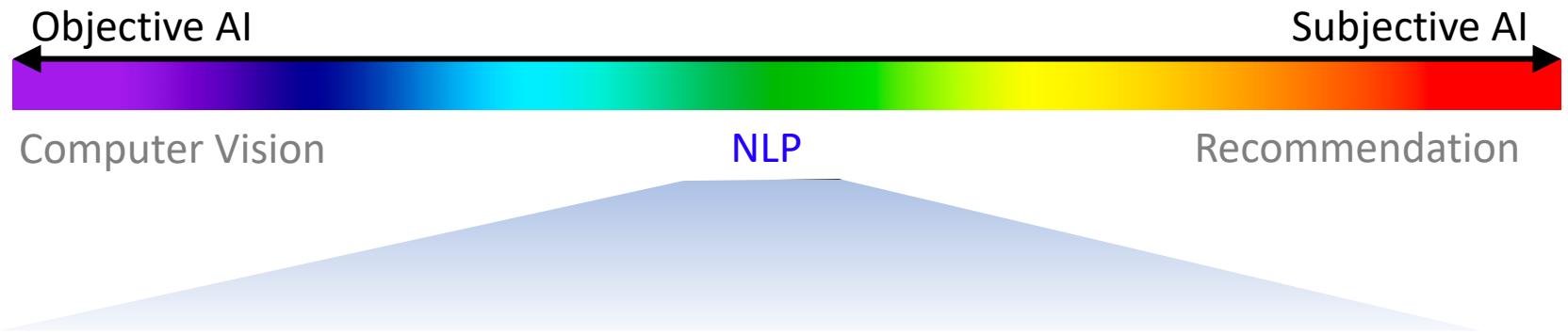
- Recommendation is **unique** in the AI family
  - Recommendation is most **close to human** among all AI tasks
  - Recommendation is a very representative **Subjective AI**
  - Thus, leads to many **unique challenges** in recommendation research



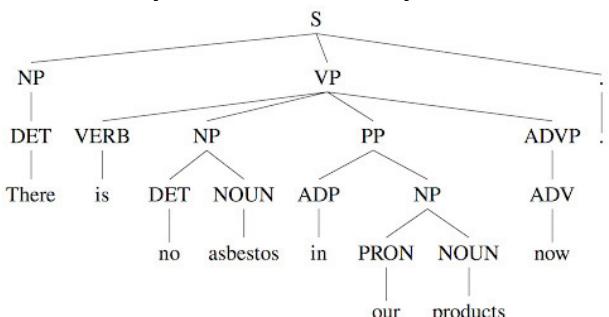
# Computer Vision: (mostly) Objective AI Tasks



# NLP: partly Objective, partly Subjective



## Syntactic Analysis



## Word Segmentation

**Words:** 这是 一篇 有趣 的 文章

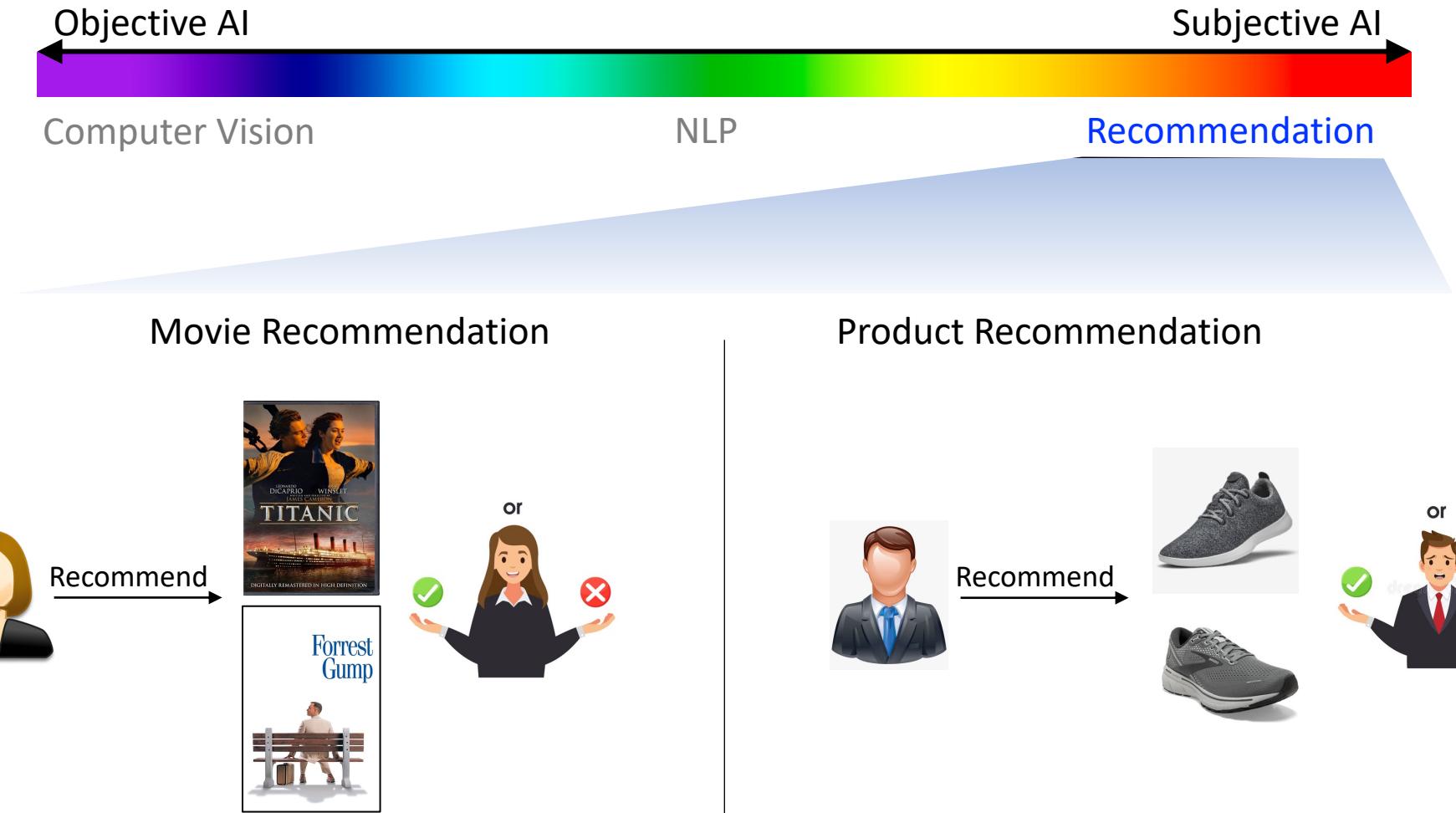
/ / / / /

(zhèshì yīpiān yǒuqù de wénzhāng)

## Dialog Systems

- Can you find me a *mobile phone* on Amazon?  
 Sure, what *operating system* do you prefer? 🤖  
 I want an *Android* one.  
 OK, and any preference on *screen size*? 🤖  
 Better larger than *5 inches*.  
 Do you have requirements on *storage capacity*? 🤖  
 I want it to be at least *64 Gigabytes*.  
 And any preference on *phone color*? 🤖  
 Not particularly.  
 Sure, then what about the following choices? 🤖
- 
- I don't like them very much...  
 OK, do you have any preference on the *brand*? 🤖  
 Better be *Samsung* or *Huawei*.  
 Any requirement on *price*? 🤖  
 Should be *within 700 dollars*.  
 OK, then what about these ones? 🤖
- 
- Great, I want the first one, can you order it for me?  
 Sure, I have placed the order for you, enjoy! 🤖

# Recommendation: mostly Subjective AI Tasks



# Recommendation is not only about Item Ranking

- A diverse set of recommendation tasks
  - Rating Prediction
  - Item Ranking
  - Sequential Recommendation
  - User Profile Construction
  - Review Summarization
  - Explanation Generation
  - Fairness Consideration
  - etc.

# Can we Handle all RecSys tasks Together?

- A diverse set of recommendation tasks
  - Rating Prediction
  - Item Ranking
  - Sequential Recommendation
  - User Profile Construction
  - Review Summarization
  - Explanation Generation
  - Fairness Consideration
  - etc.
- Do we really need to design thousands of recommendation models?
  - Difficult to integrate so many models in industry production environment

# A Bird's View of Existing RecSys

- The Multi-Stage Filtering RecSys Pipeline

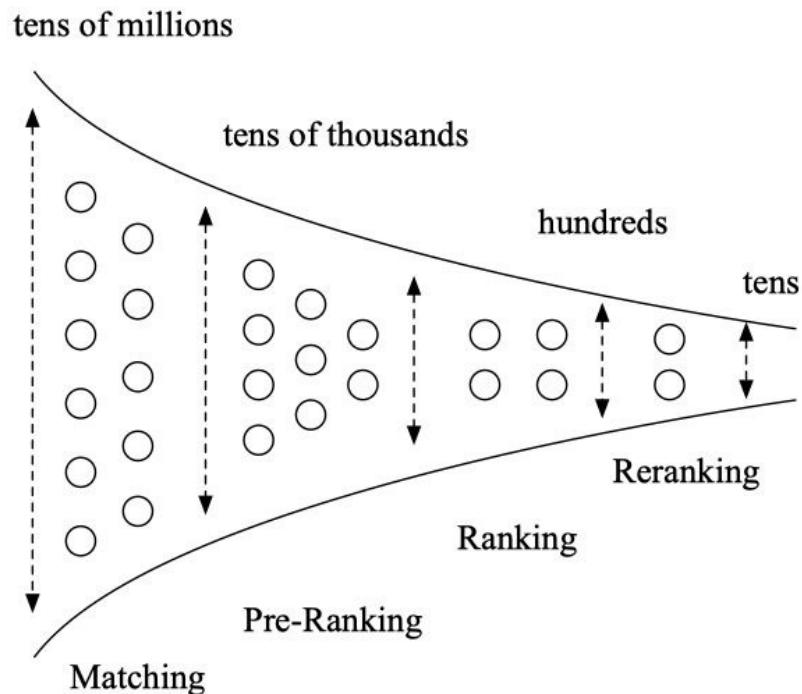


Image credit to [1]

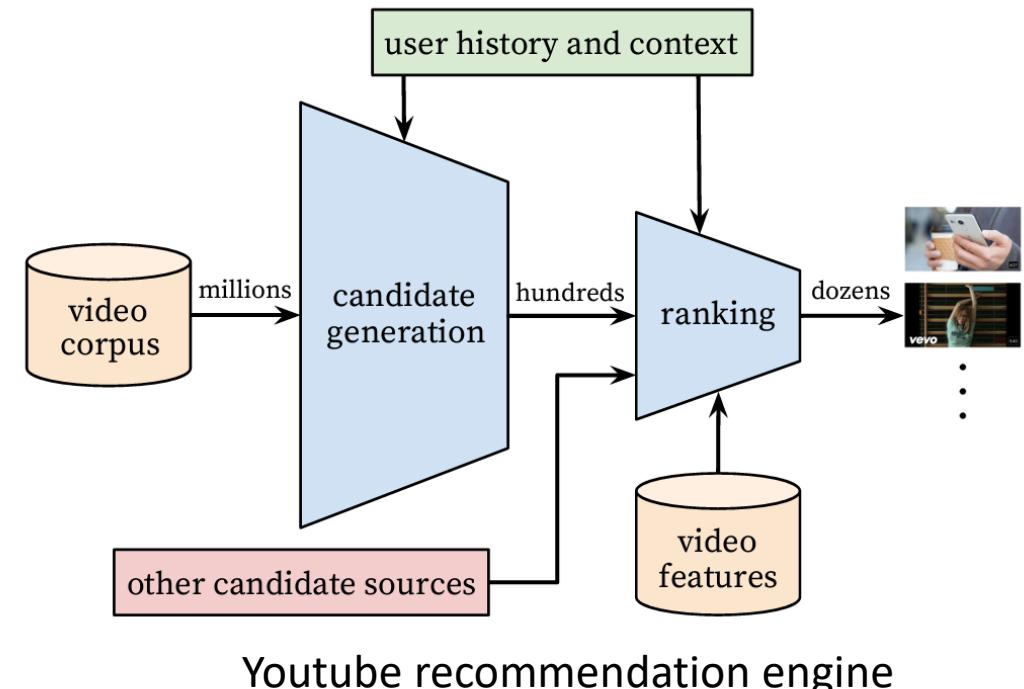


Image credit to [2]

# Discriminative Ranking

- User-item matching based on embeddings

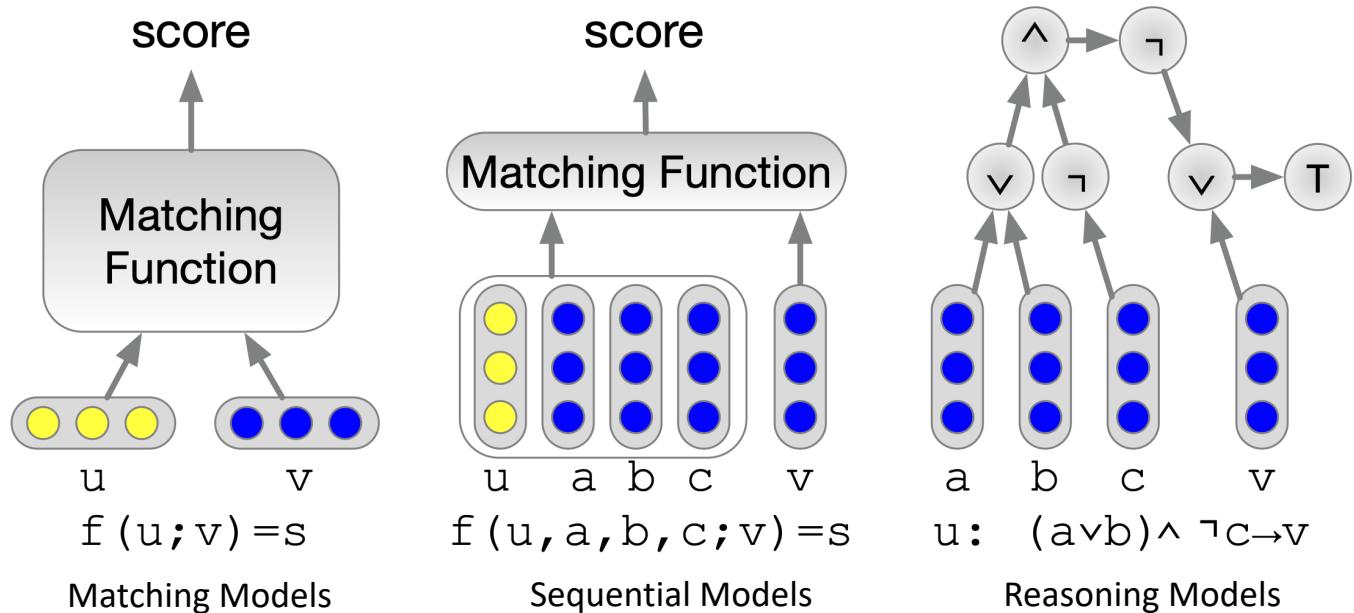


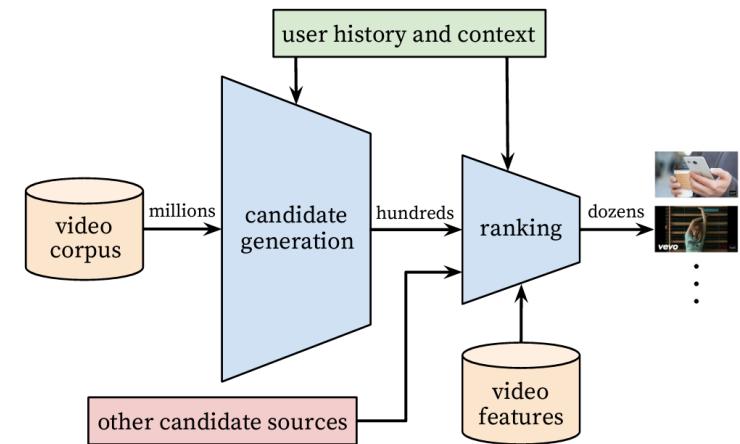
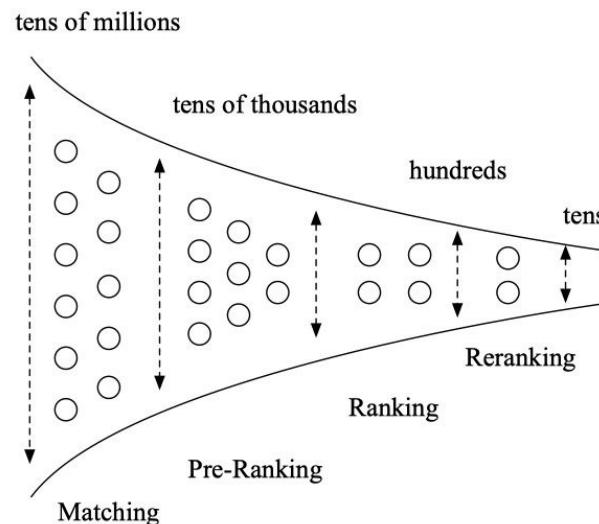
Image credit to [1]

- Discriminative ranking loss function
  - e.g., Bayesian Personalized Ranking (BPR) loss

$$\text{maximize} \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2 \quad \text{where: } \hat{x}_{uij} = p_u q_i^T - p_u q_j^T$$

# Problem with Discriminative Ranking

- Huge numbers of users and items
  - Amazon: 300 million customers, 350 million products\*
  - YouTube: 2.6+ billion monthly active users, 5+ billion videos\*\*
  - We have to use multi-stage filtering (compromise: simple rules at early stages)



- Too many candidate items, difficult for evaluation
  - Many research papers use **sampled evaluation**: 1-in-100, 1-in-1000, etc.

\*<https://sell.amazon.com/blog/amazon-stats>, and <https://www.bigcommerce.com/blog/amazon-statistics/>

\*\*<https://www.globalmediainsight.com/blog/youtube-users-statistics/>

# Large Language Models (LLMs)

- Auto-regressive decoding for generative prediction

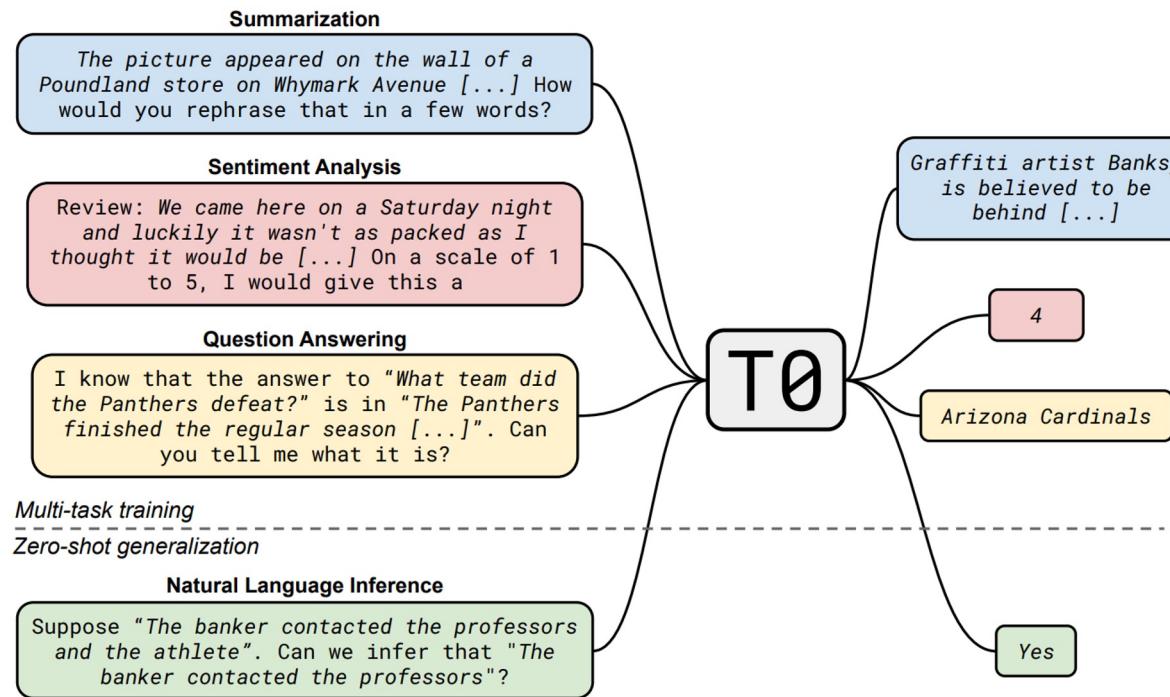


Image credit to [1]

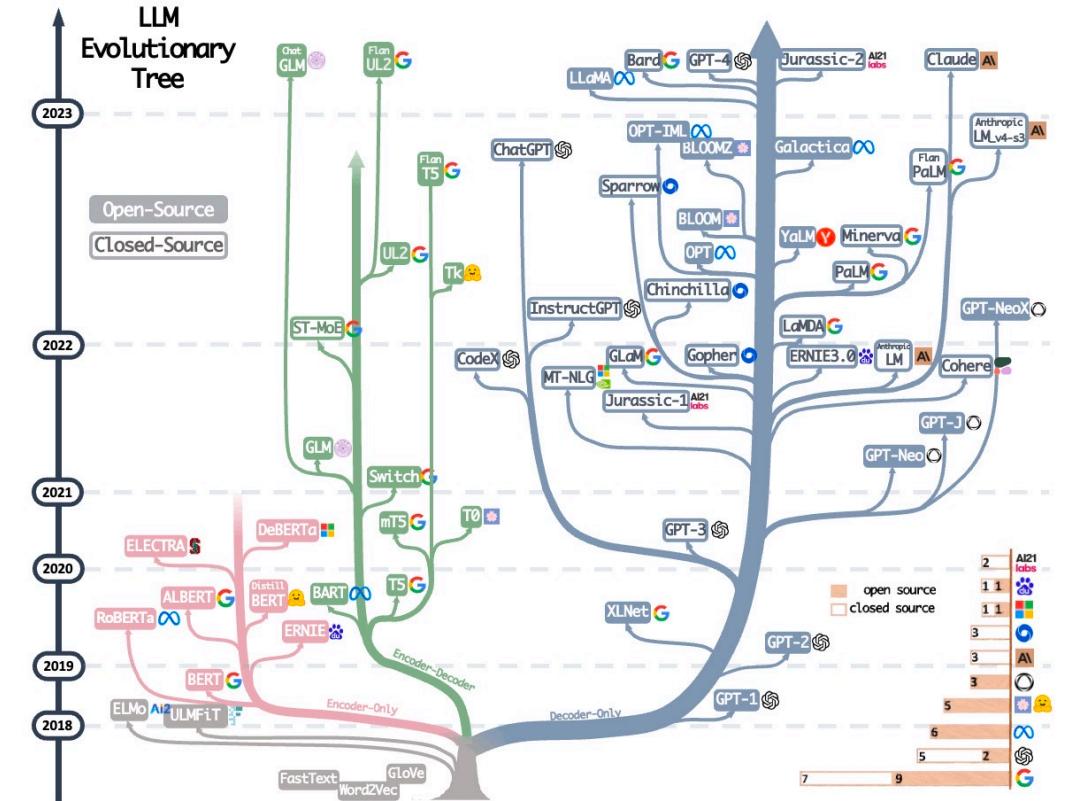


Image credit to [2]

[1] Sanh, Victor, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin et al. "Multitask prompted training enables zero-shot task generalization." ICLR 2022.

[2] Yang, Jingfeng, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond." arXiv preprint arXiv:2304.13712 (2023).

# Generative Pre-training and Prediction

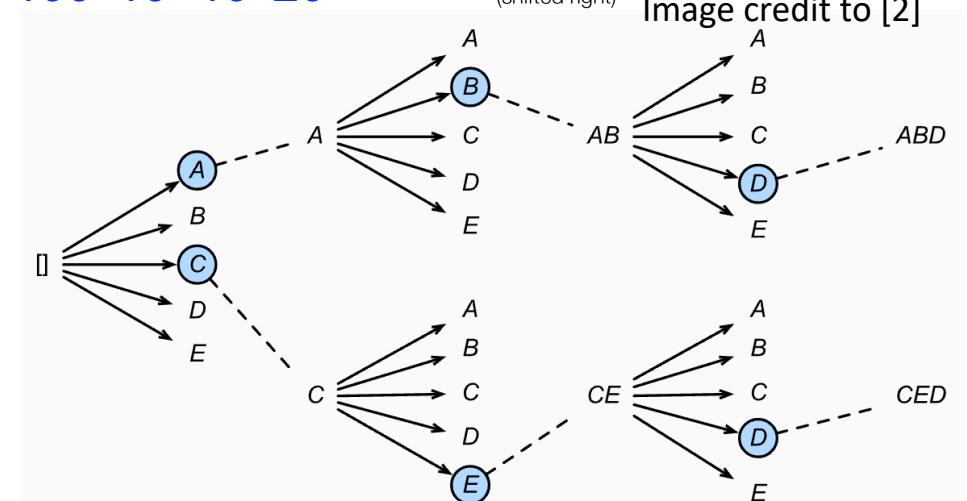
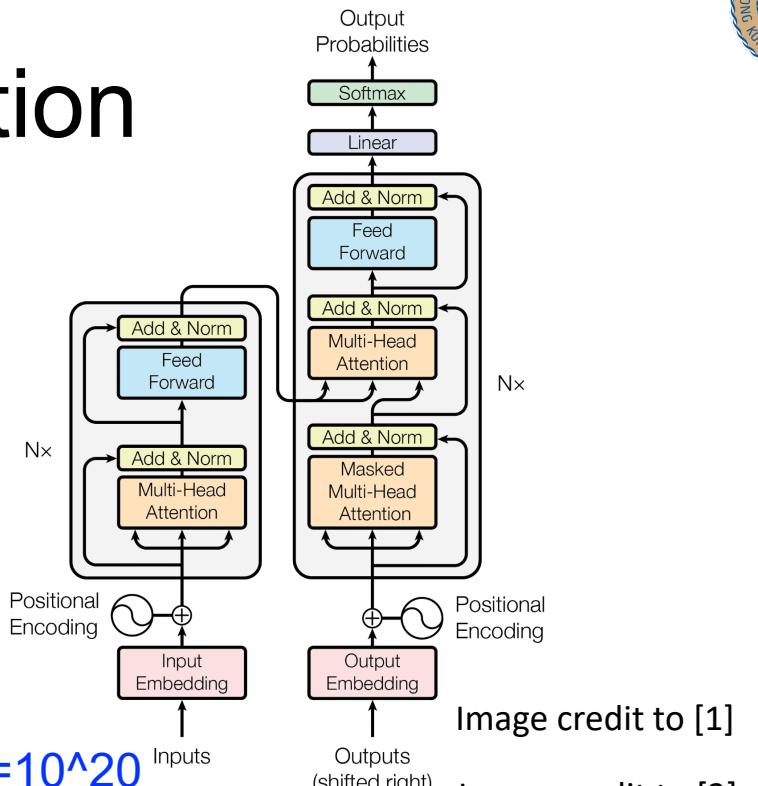
- Generative Pre-training
  - Generative Loss Function
    - Use the previous tokens to predict next token

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

- Generative Prediction

- Beam Search
  - Using finite tokens to represent infinite items
    - 100 vocabulary tokens, ID size 10 -> #items =  $100^{10} = 10^{20}$

- # of candidate tokens at each beam is fixed
- No longer need one-by-one candidate score calculation as in discriminative ranking
- Directly generate the item ID to recommend

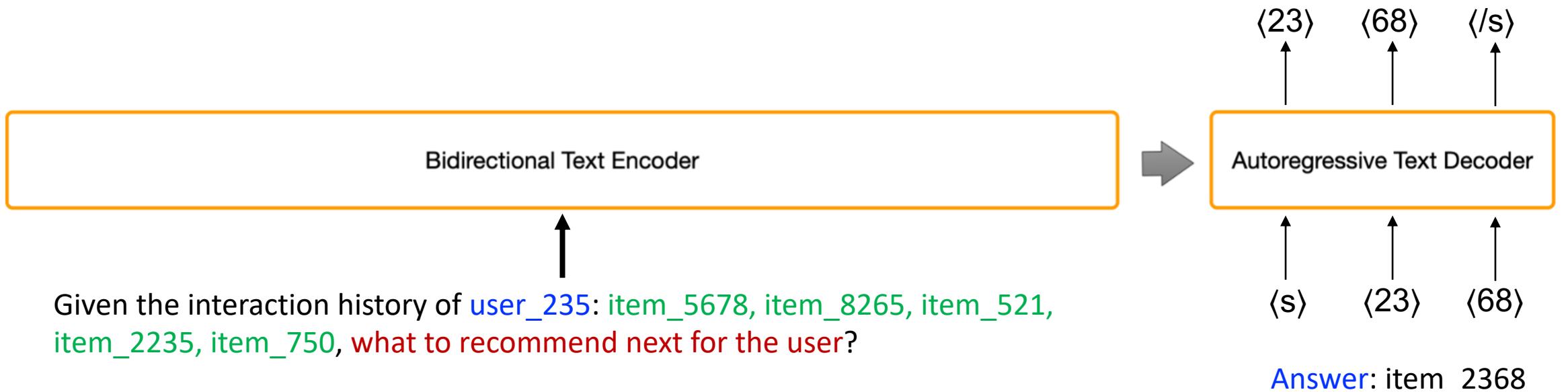


[1] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[2] [https://d2l.ai/chapter\\_recurrent-modern/beam-search.html](https://d2l.ai/chapter_recurrent-modern/beam-search.html)

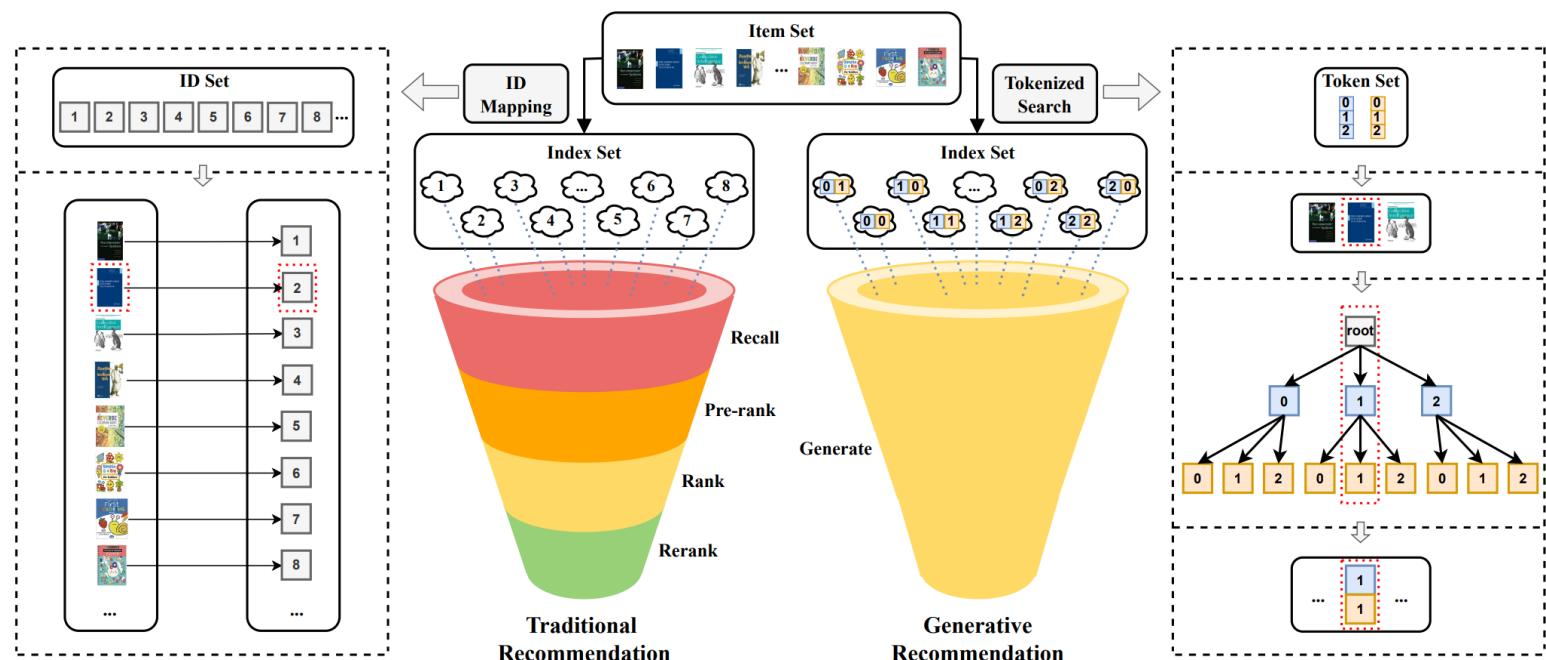
# Generative Ranking

- From Multi-stage ranking to Single-stage ranking
  - The model automatically considers **all items as the candidate pool**
  - Fixed-size item decoding
    - e.g., using 100 tokens  $\langle 00 \rangle \langle 01 \rangle \dots \langle 99 \rangle$  for item ID representation



# Generative Recommendation with Beam Search

- Since item IDs are tokenized (e.g., ["item", "\_", "73", "91"]), beam search is bounded on width
  - E.g., 100 tokens width: <00>, <01>, <02>, ..., <98>, <99>
- Assigning an item a token as in traditional recommendation is infeasible for LLM
  - Consume a lot of memory and computationally expensive



# Large Language Models for Recommendation

# How to Categorize LLM-based Recommendation

- Whether to Fine-tune LLM for Recommendation or Not
  - With Fine-tuning
  - Without Fine-tuning
- The Role of LLM in Recommendation [1]
  - LLM as RS
  - LLM in RS [2]
    - Feature Encoder
    - Ranking Function
    - ...
- Typical Recommendation Tasks [3]
  - Rating Prediction
  - Sequential Recommendation
  - ...

[1] Wu, Likang, et al. "A Survey on Large Language Models for Recommendation." *arXiv preprint arXiv:2305.19860* (2023).

[2] Lin, Jianghao, et al. "How Can Recommender Systems Benefit from Large Language Models: A Survey." *arXiv preprint arXiv:2306.05817* (2023).

[3] Li, Lei, Yongfeng Zhang, Dugang Liu, and Li Chen. "Large Language Models for Generative Recommendation: A Survey and Visionary Discussions." *arXiv preprint arXiv:2309.01157* (2023).

# Two Broad Categories of Recommendation Tasks

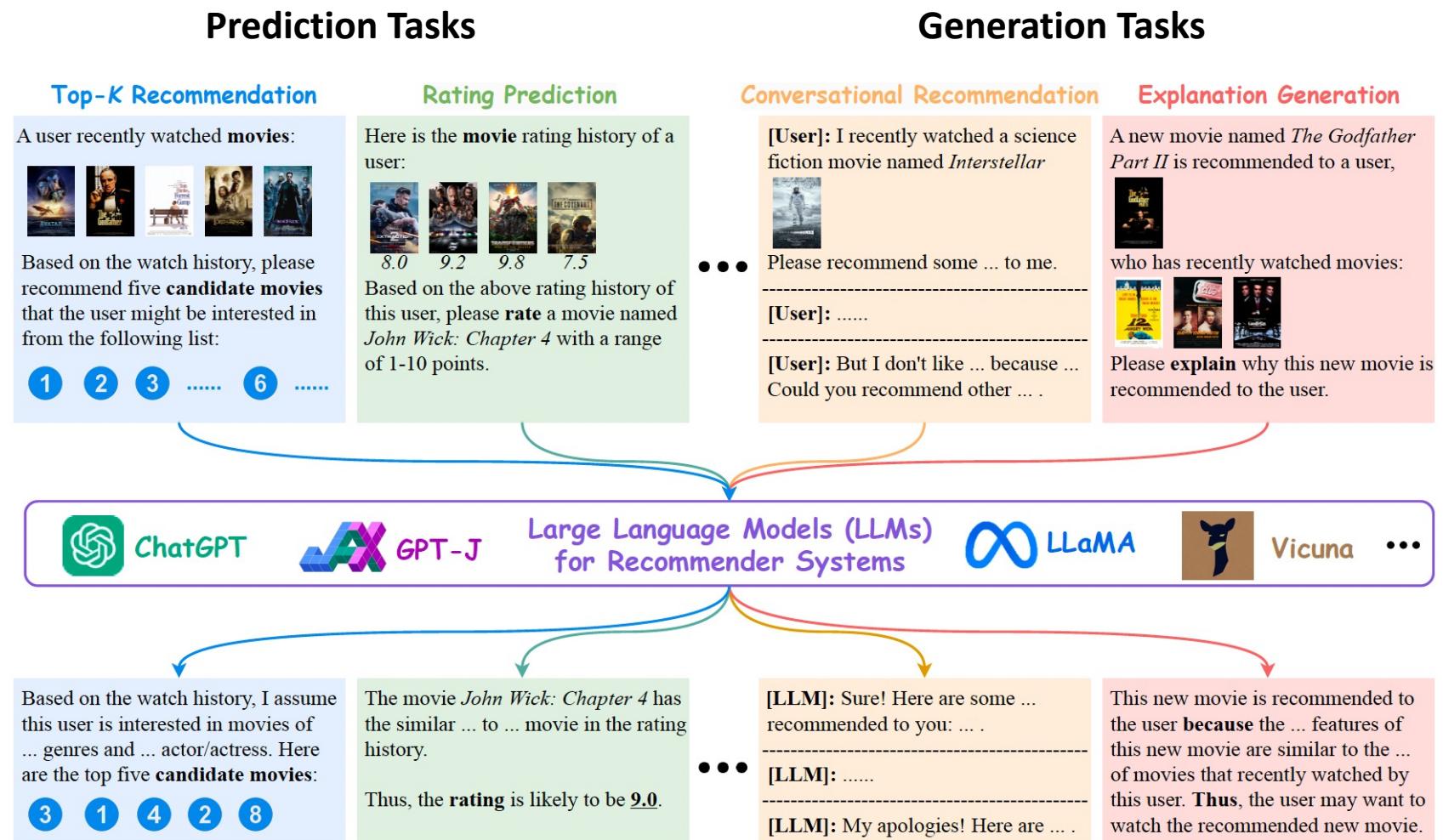


Image credit to [1]

# Typical Recommendation Tasks

- Some models can perform multiple recommendation tasks
  - e.g., P5 [2], POD [3], InstructRec [4]
- Some are task-specific
  - e.g., LMRecSys [5]

Rating Prediction	Top-N Recommendation	Sequential Recommendation	Explainable Recommendation	Review Generation	Review Summarization	Conversational Recommendation
P5 [Geng <i>et al.</i> , 2022c], BookGPT [Zhiyuli <i>et al.</i> , 2023], LLMRec [Liu <i>et al.</i> , 2023b], RecMind [Wang <i>et al.</i> , 2023b], [Liu <i>et al.</i> , 2023a; Dai <i>et al.</i> , 2023]	P5 [Geng <i>et al.</i> , 2022c], UP5 [Hua <i>et al.</i> , 2023a], VIP5 [Geng <i>et al.</i> , 2023], OpenP5 [Xu <i>et al.</i> , 2023], POD [Li <i>et al.</i> , 2023c], GPTRec [Petrov and Macdonald, 2023], LLMRec [Liu <i>et al.</i> , 2023b], RecMind [Wang <i>et al.</i> , 2023b], [Zhang <i>et al.</i> , 2023a; Zhang <i>et al.</i> , 2023b; Liu <i>et al.</i> , 2023a; Li <i>et al.</i> , 2023f; Dai <i>et al.</i> , 2023]	P5 [Geng <i>et al.</i> , 2022c], UP5 [Hua <i>et al.</i> , 2023a], VIP5 [Geng <i>et al.</i> , 2023], OpenP5 [Xu <i>et al.</i> , 2023], POD [Li <i>et al.</i> , 2023c], GenRec [Ji <i>et al.</i> , 2023], GPTRec [Petrov and Macdonald, 2023], LMRRecSys [Zhang <i>et al.</i> , 2021], NIR [Wang and Lim, 2023], PALR [Chen, 2023], LLMRec [Liu <i>et al.</i> , 2023b], RecMind [Wang <i>et al.</i> , 2023b], BIGRec [Bao <i>et al.</i> , 2023a], [Hua <i>et al.</i> , 2023b; Liu <i>et al.</i> , 2023a; Hou <i>et al.</i> , 2023b; Zhang <i>et al.</i> , 2023b]	P5 [Geng <i>et al.</i> , 2022c], VIP5 [Geng <i>et al.</i> , 2023], POD [Li <i>et al.</i> , 2023c], PEPLER [Li <i>et al.</i> , 2023b], M6-Rec [Cui <i>et al.</i> , 2022], LLMRec [Liu <i>et al.</i> , 2023b], RecMind [Wang <i>et al.</i> , 2023b], KnowRec [Colas <i>et al.</i> , 2023], [Liu <i>et al.</i> , 2023a]	-	P5 [Geng <i>et al.</i> , 2022c], LLMRec [Liu <i>et al.</i> , 2023b], RecMind [Wang <i>et al.</i> , 2023b], [Liu <i>et al.</i> , 2023a]	M6-Rec [Cui <i>et al.</i> , 2022], RecMind [Friedman <i>et al.</i> , 2023], Chat-REC [Gao <i>et al.</i> , 2023], [Wang <i>et al.</i> , 2023a; Lin and Zhang, 2023; He <i>et al.</i> , 2023]

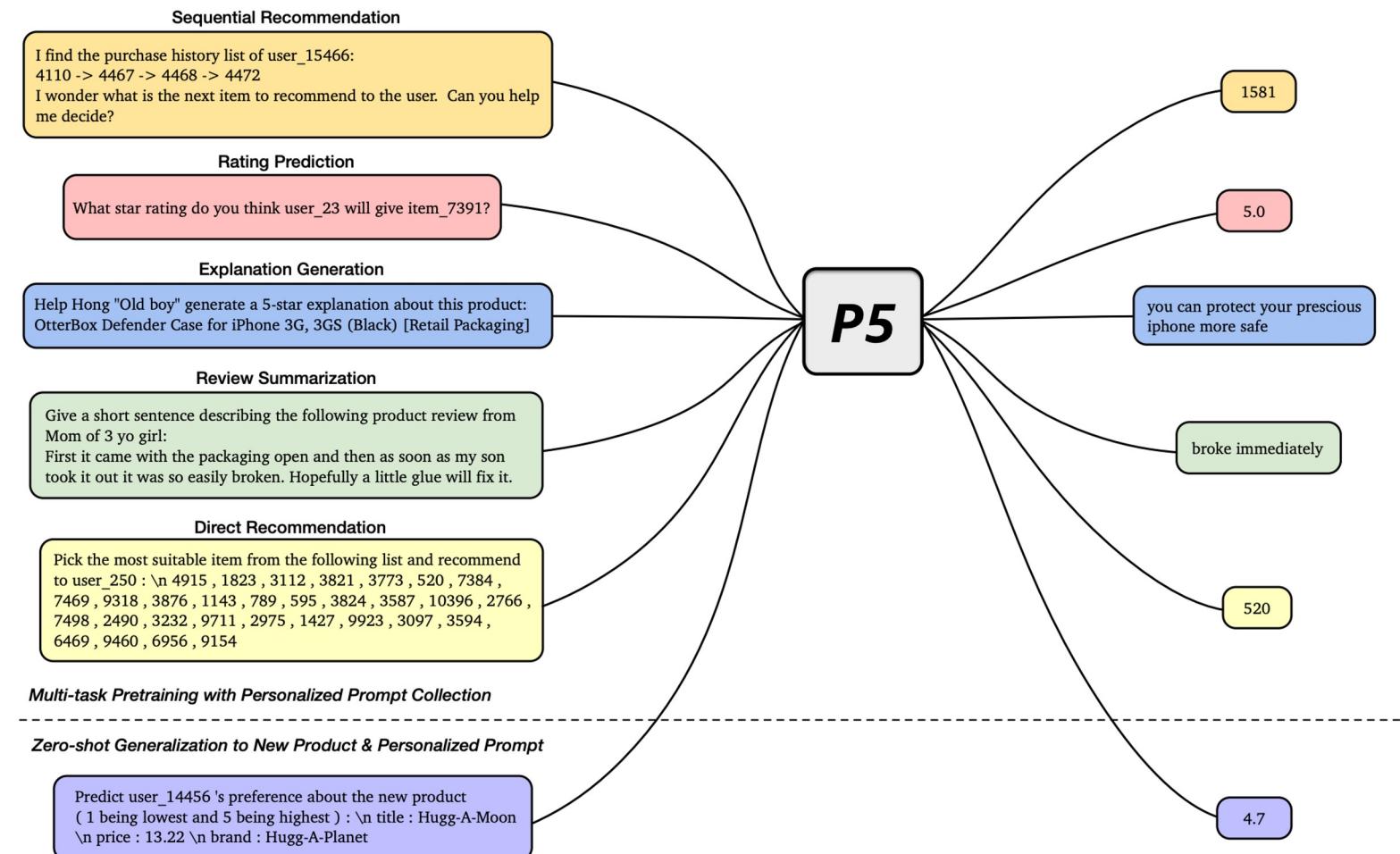
Image credit to [1]

- [1] Li, Lei, Yongfeng Zhang, Dugang Liu, and Li Chen. "Large Language Models for Generative Recommendation: A Survey and Visionary Discussions." *arXiv preprint arXiv:2309.01157* (2023).
- [2] Geng, Shijie, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)." *RecSys 2022*.
- [3] Li, Lei, Yongfeng Zhang, and Li Chen. "Prompt Distillation for Efficient LLM-based Recommendation." *CIKM 2023*.
- [4] Zhang, Junjie, et al. "Recommendation as instruction following: A large language model empowered recommendation approach." *arXiv preprint arXiv:2305.07001* (2023).
- [5] Zhang, Yuhui, et al. "Language models as recommender systems: Evaluations and limitations." *NeurIPS 2021 Workshop ICBINB*, 2021.

# The P5 Generative Recommendation Paradigm

- P5: Pretrain, Personalized Prompt & Predict Paradigm [1]

- Learns **multiple** recommendation tasks together through a unified **sequence-to-sequence** framework
- Formulates different recommendation problems as **prompt-based natural language tasks**
- User-item information and corresponding features are integrated with **personalized prompts** as model inputs



# Personalization in Prompts

- Definition of **personalized prompts**
  - A prompt that includes personalized fields for different users and items
- User's preference can be indicated through
  - A **user ID** (e.g., "user\_23")
  - Content **description of the user** such as location, preferred movie genres, etc.
- Item field can be represented by
  - An **item ID** (e.g., "item\_7391")
  - Item **content metadata** that contains **detailed descriptions** of the item, e.g., item category

# Personalized Prompt Design

## Rating / Review / Explanation raw data for Beauty

```

user_id: 7641      user_name: stephanie
item_id: 2051
item_title: SHANY Nail Art Set (24 Famouse Colors
Nail Art Polish, Nail Art Decoration)
review: Absolutely great product. I bought this for my fourteen year
old niece for Christmas and of course I had to try it out, then I
tried another one, and another one and another one. So much fun!
I even contemplated keeping a few for myself!
star_rating: 5
summary: Perfect!
explanation: Absolutely great product      feature_word: product
  
```

(a)

Which star rating will user\_{{user\_id}} give item\_{{item\_id}}?  
(1 being lowest and 5 being highest) → {{star\_rating}}

Based on the feature word {{feature\_word}}, generate an explanation for user\_{{user\_id}} about this product:  
<{{item\_title}>} → {{explanation}}

Give a short sentence describing the following product review  
from {{user\_name}}: {{review}} → {{summary}}

## Sequential Recommendation raw data for Beauty

```

user_id: 7641      user_name: Victor
purchase_history: 652 -> 460 -> 447 -> 653 -> 654 -> 655 -> 656 -> 8
-> 657
next_item: 552
candidate_items: 4885 , 4280 , 4886 , 1907 , 870 , 4281 , 4222 ,
4887 , 2892 , 4888 , 2879 , 3147 , 2195 , 3148 , 3179 , 1951 ,
..... , 1982 , 552 , 2754 , 2481 , 1916 , 2822 , 1325
  
```

(b)

Here is the purchase history of user\_{{user\_id}}:  
<{{purchase\_history}>  
What to recommend next for the user? → {{next\_item}}

## Direct Recommendation raw data for Beauty

```

user_id: 250      user_name: moriah rose
target_item: 520
random_negative_item: 9711
candidate_items: 4915 , 1823 , 3112 , 3821 , 3773 , 520 , 7384 ,
7469 , 9318 , 3876 , 1143 , 789 , 595 , 3824 , 3587 , 10396 ,
..... , 2766 , 7498 , 2490 , 3232 , 9711 , 2975 , 1405 , 8051
  
```

(c)

Choose the best item from the candidates to recommend for  
<{{user\_name}}>? \n <{{candidate\_items}}> → {{target\_item}}

# Design Multiple Prompts for Each Task

- To enhance variation in language style (e.g., sequential recommendation)

## Prompt ID: 2-1

Input template: Given the following purchase history of user\_{{user\_id}}:  
{{purchase\_history}}  
predict next possible item to be purchased by the user?

Target template: {{next\_item}}

## Prompt ID: 2-2

Input template: I find the purchase history list of user\_{{user\_id}}:  
{{purchase\_history}}  
I wonder which is the next item to recommend to the user. Can you help me decide?

Target template: {{next\_item}}

## Prompt ID: 2-3

Input template: Here is the purchase history list of user\_{{user\_id}}:  
{{purchase\_history}}  
try to recommend next item to the user

Target template: {{next\_item}}

## Prompt ID: 2-4

Input template: Given the following purchase history of {{user\_desc}}:  
{{purchase\_history}}  
predict next possible item for the user

Target template: {{next\_item}}

## Prompt ID: 2-5

Input template: Based on the purchase history of {{user\_desc}}:  
{{purchase\_history}}  
Can you decide the next item likely to be purchased by the user?

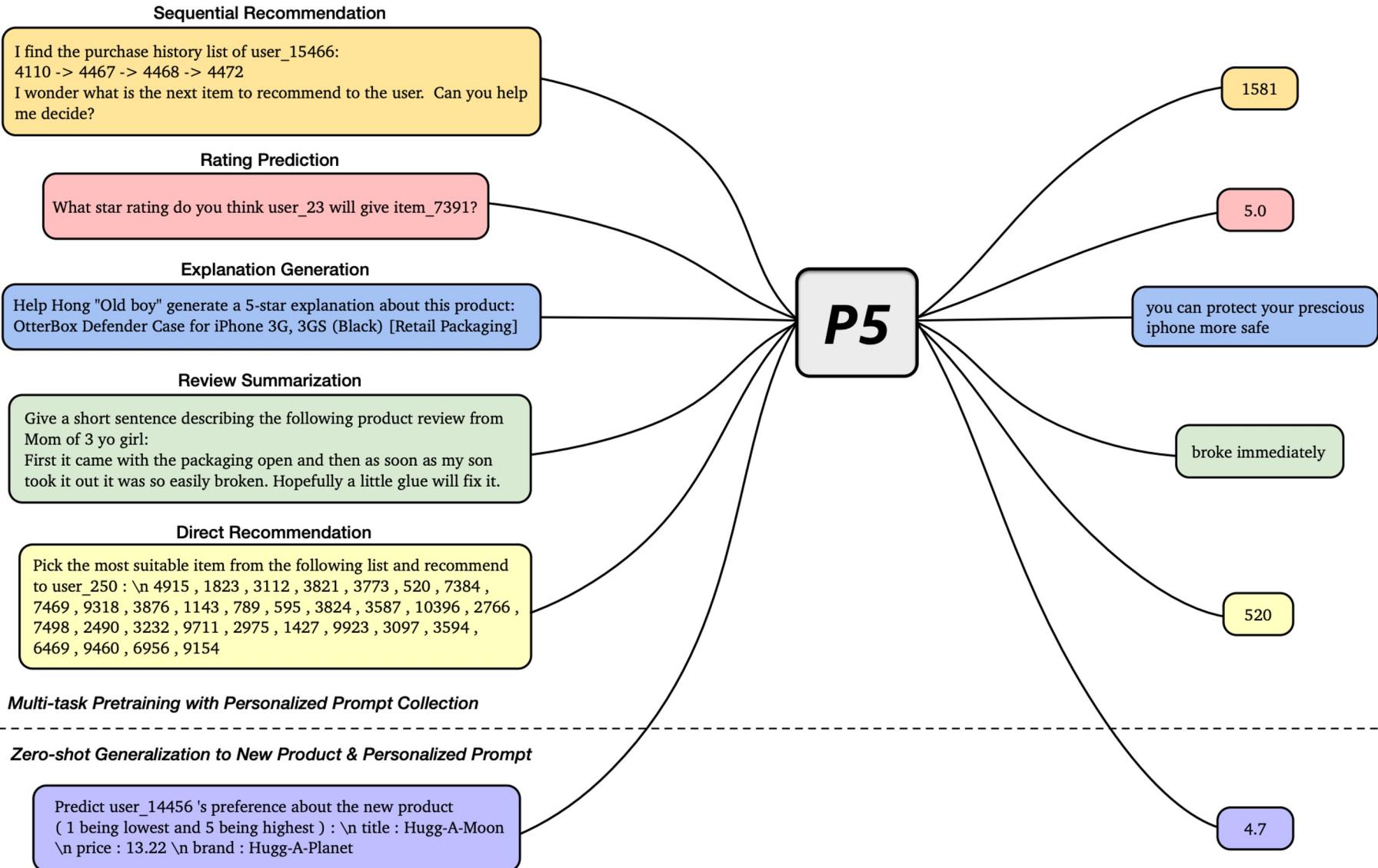
Target template: {{next\_item}}

## Prompt ID: 2-6

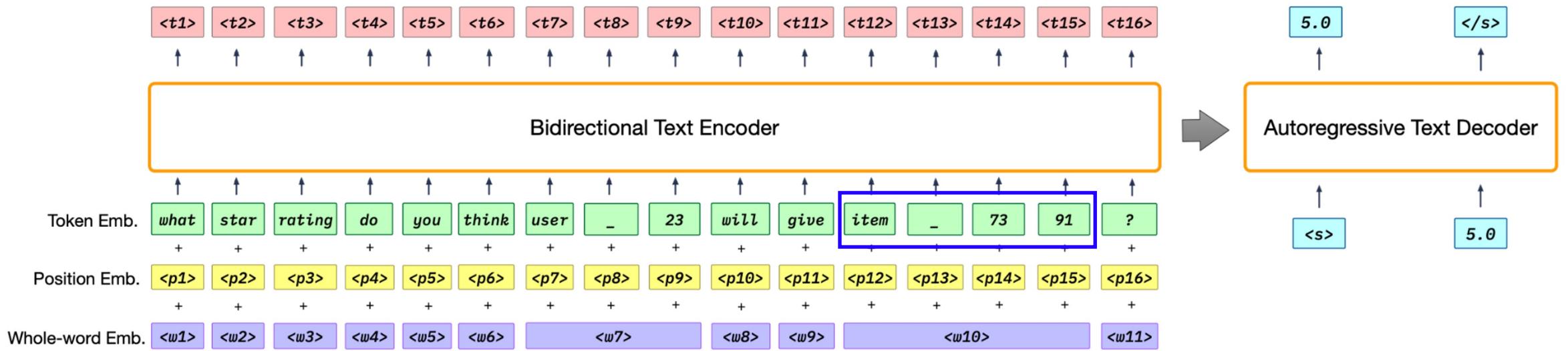
Input template: Here is the purchase history of {{user\_desc}}:  
{{purchase\_history}}  
What to recommend next for the user?

Target template: {{next\_item}}

# Multi-Task Pre-training



# Multi-Task Pre-training



- P5 is pre-trained on top of T5 checkpoints (to enable basic ability for language understanding)
- By default, P5 uses **multiple sub-word units** to represent personalized fields (e.g., ["item", "\_", "73", "91"])

# Generative Recommendation

- The encoder takes input sequence
- The decoder autoregressively generates next words:

◦ **Autoregressive LM loss** is shared by all tasks:  $\mathcal{L}_{\theta}^{\text{P5}} = - \sum_{j=1}^{|\mathbf{y}|} \log P_{\theta}(\mathbf{y}_j | \mathbf{y}_{<j}, \mathbf{x})$

- P5 can unify various recommendation tasks with **one model, one loss, and one data format**

- Inference with pretrained P5
  - Simply apply **beam search** to generate a list of potential next items
    - Beam size set to N  $\rightarrow$  N candidates

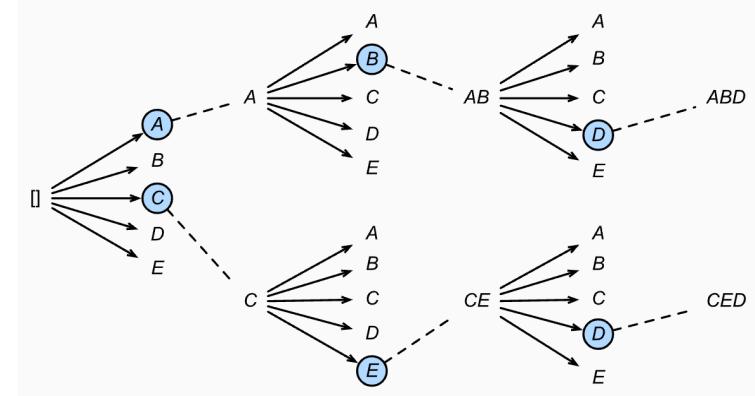


Image credit to [1]

# Advantages of P5 Generative Recommendation

- Immerses recommendation models into a full language environment
  - With the **flexibility** and **expressiveness** of language, there is **no need to design feature-specific encoders**
- P5 treats all personalized tasks as a conditional text generation problem
  - One data format, one model, one loss for multiple recommendation tasks
  - No need to design data-specific or task-specific **recommendation models**
- P5 attains sufficient **zero-shot performance** when generalizing to novel personalized prompts or unseen items in other domains

# Performance of P5 under **seen** Prompts

## Rating Prediction:

Methods	Sports		Beauty		Toys	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
MF	<b>1.0234</b>	0.7935	<b>1.1973</b>	0.9461	<b>1.0123</b>	0.7984
MLP	1.1277	0.7626	1.3078	0.9597	1.1215	0.8097
P5-S (1-6)	1.0594	<b>0.6639</b>	1.3128	<b>0.8428</b>	1.0746	<b>0.7054</b>
P5-B (1-6)	1.0357	0.6813	<u>1.2843</u>	0.8534	1.0544	0.7177
P5-S (1-10)	1.0522	<u>0.6698</u>	1.2989	<u>0.8473</u>	1.0550	0.7173
P5-B (1-10)	<u>1.0292</u>	0.6864	1.2870	0.8531	<u>1.0245</u>	<b>0.6931</b>

## Sequential Recommendation:

Methods	Sports				Beauty				Toys			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0166	0.0107	0.0270	0.0141
HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0321	0.0221	0.0497	0.0277
GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0097	0.0059	0.0176	0.0084
BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0116	0.0071	0.0203	0.0099
FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0228	0.0140	0.0381	0.0189
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0463	0.0306	0.0675	0.0374
S <sup>3</sup> -Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0443	0.0294	0.0700	0.0376
P5-S (2-3)	0.0272	0.0169	0.0361	0.0198	<u>0.0503</u>	<u>0.0370</u>	<u>0.0659</u>	<u>0.0421</u>	<b>0.0648</b>	<b>0.0567</b>	<b>0.0709</b>	<b>0.0587</b>
P5-B (2-3)	<u>0.0364</u>	<u>0.0296</u>	<u>0.0431</u>	<u>0.0318</u>	<b>0.0508</b>	<b>0.0379</b>	<b>0.0664</b>	<b>0.0429</b>	0.0608	0.0507	0.0688	0.0534
P5-S (2-13)	0.0258	0.0159	0.0346	0.0188	0.0490	0.0358	0.0646	0.0409	<u>0.0647</u>	<u>0.0566</u>	<u>0.0705</u>	<u>0.0585</u>
P5-B (2-13)	<b>0.0387</b>	<b>0.0312</b>	<b>0.0460</b>	<b>0.0336</b>	0.0493	0.0367	0.0645	0.0416	0.0587	0.0486	0.0675	0.0536

## Explanation Generation:

Methods	Sports				Beauty				Toys			
	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL
Attn2Seq	0.5305	12.2800	1.2107	9.1312	0.7889	12.6590	1.6820	9.7481	1.6238	13.2245	2.9942	10.7398
NRT	0.4793	11.0723	1.1304	7.6674	0.8295	12.7815	1.8543	9.9477	1.9084	13.5231	3.6708	11.1867
PETER	0.7112	12.8944	1.3283	9.8635	<u>1.1541</u>	14.8497	<u>2.1413</u>	11.4143	1.9861	14.2716	3.6718	11.7010
P5-S (3-3)	<b>1.0447</b>	<b>14.9048</b>	<b>2.1297</b>	<b>11.1778</b>	<b>1.2237</b>	<b>17.6938</b>	<b>2.2489</b>	<b>12.8606</b>	<u>2.2892</u>	<b>15.4505</b>	<u>3.6974</u>	<b>12.1718</b>
P5-B (3-3)	<u>1.0407</u>	<u>14.1589</u>	<u>2.1220</u>	<u>10.6096</u>	0.9742	<u>16.4530</u>	1.8858	<u>11.8765</u>	<b>2.3185</b>	<u>15.3474</u>	<b>3.7209</b>	<u>12.1312</u>
PETER+	<b>2.4627</b>	<b>24.1181</b>	5.1937	<b>18.4105</b>	<b>3.2606</b>	<u>25.5541</u>	5.9668	<u>19.7168</u>	<b>4.7919</b>	<u>28.3083</u>	9.4520	<b>22.7017</b>
P5-S (3-9)	1.4101	<u>23.5619</u>	<b>5.4196</b>	<u>17.6245</u>	<u>1.9788</u>	<b>25.6253</b>	<b>6.3678</b>	<b>19.9497</b>	4.1222	<b>28.4088</b>	9.5432	<u>22.6064</u>
P5-B (3-9)	<u>1.4689</u>	23.5476	<u>5.3926</u>	17.5852	1.8765	25.1183	6.0764	19.4488	3.8933	27.9916	<u>9.5896</u>	22.2178
P5-S (3-12)	1.3212	23.2474	5.3461	17.3780	1.9425	25.1474	6.0551	19.5601	<u>4.2764</u>	28.1897	9.1327	22.2514
P5-B (3-12)	1.4303	23.3810	5.3239	17.4913	1.9031	25.1763	<u>6.1980</u>	19.5188	3.5861	28.1369	<b>9.7562</b>	22.3056

# Performance of P5 under **seen** Prompts

## Review Summarization:

Methods	Sports				Beauty				Toys			
	BLUE2	ROUGE1	ROUGE2	ROUGEL	BLUE2	ROUGE1	ROUGE2	ROUGEL	BLUE2	ROUGE1	ROUGE2	ROUGEL
T0 <b>(4-1)</b>	2.1581	2.2695	0.5694	1.6221	1.2871	1.2750	0.3904	0.9592	<u>2.2296</u>	2.4671	0.6482	1.8424
GPT-2 <b>(4-1)</b>	0.7779	4.4534	1.0033	1.9236	0.5879	3.3844	0.6756	1.3956	0.6221	3.7149	0.6629	1.4813
P5-S <b>(4-1)</b>	<u>2.4962</u>	<u>11.6701</u>	<u>2.7187</u>	<u>10.4819</u>	<b>2.1225</b>	<b>8.4205</b>	<b>1.6676</b>	<b>7.5476</b>	<b>2.4752</b>	<b>9.4200</b>	<b>1.5975</b>	<b>8.2618</b>
P5-B <b>(4-1)</b>	<b>2.6910</b>	<b>12.0314</b>	<b>3.2921</b>	<b>10.7274</b>	<u>1.9325</u>	<u>8.2909</u>	<u>1.4321</u>	<u>7.4000</u>	1.7833	<u>8.7222</u>	<u>1.3210</u>	<u>7.6134</u>

## Direct Recommendation:

Methods	Sports					Beauty					Toys				
	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10
BPR-MF	0.0314	0.1404	0.0848	0.2563	0.1220	0.0311	0.1426	0.0857	0.2573	0.1224	0.0233	0.1066	0.0641	0.2003	0.0940
BPR-MLP	0.0351	0.1520	0.0927	0.2671	0.1296	0.0317	0.1392	0.0848	0.2542	0.1215	0.0252	0.1142	0.0688	0.2077	0.0988
SimpleX	0.0331	<b>0.2362</b>	<b>0.1505</b>	<u>0.3290</u>	<u>0.1800</u>	0.0325	<u>0.2247</u>	<u>0.1441</u>	0.3090	<u>0.1711</u>	0.0268	<b>0.1958</b>	<b>0.1244</b>	<b>0.2662</b>	<b>0.1469</b>
P5-S <b>(5-1)</b>	0.0638	0.2096	0.1375	0.3143	0.1711	0.0600	0.2021	0.1316	<u>0.3121</u>	0.1670	0.0405	<u>0.1538</u>	<u>0.0969</u>	<u>0.2405</u>	<u>0.1248</u>
P5-B <b>(5-1)</b>	0.0245	0.0816	0.0529	0.1384	0.0711	0.0224	0.0904	0.0559	0.1593	0.0780	0.0187	0.0827	0.0500	0.1543	0.0729
P5-S <b>(5-4)</b>	<u>0.0701</u>	<u>0.2241</u>	<u>0.1483</u>	<b>0.3313</b>	<b>0.1827</b>	<b>0.0862</b>	<b>0.2448</b>	<b>0.1673</b>	<b>0.3441</b>	<b>0.1993</b>	0.0413	0.1411	0.0916	0.2227	0.1178
P5-B <b>(5-4)</b>	0.0299	0.1026	0.0665	0.1708	0.0883	0.0506	0.1557	0.1033	0.2350	0.1287	0.0435	0.1316	0.0882	0.2000	0.1102
P5-S <b>(5-5)</b>	0.0574	0.1503	0.1050	0.2207	0.1276	0.0601	0.1611	0.1117	0.2370	0.1360	<u>0.0440</u>	0.1282	0.0865	0.2011	0.1098
P5-B <b>(5-5)</b>	0.0641	0.1794	0.1229	0.2598	0.1488	0.0588	0.1573	0.1089	0.2325	0.1330	0.0386	0.1122	0.0756	0.1807	0.0975
P5-S <b>(5-8)</b>	0.0567	0.1514	0.1049	0.2196	0.1269	0.0571	0.1566	0.1078	0.2317	0.1318	<b>0.0451</b>	0.1322	0.0889	0.2023	0.1114
P5-B <b>(5-8)</b>	<b>0.0726</b>	0.1955	0.1355	0.2802	0.1627	<u>0.0608</u>	0.1564	0.1096	0.2300	0.1332	0.0389	0.1147	0.0767	0.1863	0.0997

Observation: P5 **achieves promising performances** on the five task families when taking **seen prompt** templates as model inputs

# Performance of P5 under **unseen** Prompts

**Observation:** Multitask prompted pretraining empowers P5 **good robustness** to understand **unseen prompts** with wording variations

Sequential Recommendation:

Methods	Sports				Beauty				Toys			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0166	0.0107	0.0270	0.0141
HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0321	0.0221	0.0497	0.0277
GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0097	0.0059	0.0176	0.0084
BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0116	0.0071	0.0203	0.0099
FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0228	0.0140	0.0381	0.0189
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0463	0.0306	0.0675	0.0374
S <sup>3</sup> -Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0443	0.0294	0.0700	0.0376
P5-S (2-3)	0.0272	0.0169	0.0361	0.0198	0.0503	0.0370	0.0659	0.0421	<b>0.0648</b>	<b>0.0567</b>	<b>0.0709</b>	<b>0.0587</b>
P5-B (2-3)	<b>0.0364</b>	<b>0.0296</b>	<b>0.0431</b>	<b>0.0318</b>	<b>0.0508</b>	<b>0.0379</b>	<b>0.0664</b>	<b>0.0429</b>	0.0608	0.0507	0.0688	0.0534
P5-S (2-13)	0.0258	0.0159	0.0346	0.0188	0.0490	0.0358	0.0646	0.0409	0.0647	0.0566	0.0705	0.0585
P5-B (2-13)	<b>0.0387</b>	<b>0.0312</b>	<b>0.0460</b>	<b>0.0336</b>	0.0493	0.0367	0.0645	0.0416	0.0587	0.0486	0.0675	0.0536

Explanation Generation:

Methods	Sports				Beauty				Toys			
	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL
Attn2Seq	0.5305	12.2800	1.2107	9.1312	0.7889	12.6590	1.6820	9.7481	1.6238	13.2245	2.9942	10.7398
NRT	0.4793	11.0723	1.1304	7.6674	0.8295	12.7815	1.8543	9.9477	1.9084	13.5231	3.6708	11.1867
PETER	0.7112	12.8944	1.3283	9.8635	<u>1.1541</u>	14.8497	<u>2.1413</u>	11.4143	1.9861	14.2716	3.6718	11.7010
P5-S (3-3)	<b>1.0447</b>	<b>14.9048</b>	<b>2.1297</b>	<b>11.1778</b>	<b>1.2237</b>	<b>17.6938</b>	<b>2.2489</b>	<b>12.8606</b>	<u>2.2892</u>	<b>15.4505</b>	<u>3.6974</u>	<b>12.1718</b>
P5-B (3-3)	<u>1.0407</u>	<u>14.1589</u>	<u>2.1220</u>	<u>10.6096</u>	0.9742	<u>16.4530</u>	1.8858	<u>11.8765</u>	<b>2.3185</b>	<u>15.3474</u>	<b>3.7209</b>	<u>12.1312</u>
PETER+	<b>2.4627</b>	<b>24.1181</b>	5.1937	<b>18.4105</b>	<b>3.2606</b>	<u>25.5541</u>	5.9668	<b>19.7168</b>	<b>4.7919</b>	<u>28.3083</u>	9.4520	<b>22.7017</b>
P5-S (3-9)	1.4101	<u>23.5619</u>	<b>5.4196</b>	<u>17.6245</u>	<u>1.9788</u>	<b>25.6253</b>	<b>6.3678</b>	<b>19.9497</b>	4.1222	<b>28.4088</b>	9.5432	<u>22.6064</u>
P5-B (3-9)	<u>1.4689</u>	23.5476	<u>5.3926</u>	17.5852	1.8765	25.1183	6.0764	19.4488	3.8933	27.9916	<u>9.5896</u>	22.2178
P5-S (3-12)	1.3212	23.2474	5.3461	17.3780	1.9425	25.1474	6.0551	19.5601	<u>4.2764</u>	28.1897	9.1327	22.2514
P5-B (3-12)	1.4303	23.3810	5.3239	17.4913	1.9031	25.1763	<u>6.1980</u>	19.5188	3.5861	28.1369	<b>9.7562</b>	22.3056

Direct Recommendation:

Methods	Sports					Beauty					Toys				
	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10
BPR-MF	0.0314	0.1404	0.0848	0.2563	0.1220	0.0311	0.1426	0.0857	0.2573	0.1224	0.0233	0.1066	0.0641	0.2003	0.0940
BPR-MLP	0.0351	0.1520	0.0927	0.2671	0.1296	0.0317	0.1392	0.0848	0.2542	0.1215	0.0252	0.1142	0.0688	0.2077	0.0988
SimpleX	0.0331	<b>0.2362</b>	<b>0.1505</b>	<u>0.3290</u>	<u>0.1800</u>	0.0325	<u>0.2247</u>	<u>0.1441</u>	0.3090	<u>0.1711</u>	0.0268	<b>0.1958</b>	<b>0.1244</b>	<b>0.2662</b>	<b>0.1469</b>
P5-S (5-1)	0.0638	0.2096	0.1375	0.3143	0.1711	0.0600	0.2021	0.1316	<u>0.3121</u>	0.1670	0.0405	<u>0.1538</u>	<u>0.0969</u>	<u>0.2405</u>	0.1248
P5-B (5-1)	0.0245	0.0816	0.0529	0.1384	0.0711	0.0224	0.0904	0.0559	0.1593	0.0780	0.0187	0.0827	0.0500	0.1543	0.0729
P5-S (5-4)	<u>0.0701</u>	<u>0.2241</u>	<u>0.1483</u>	<b>0.3313</b>	<b>0.1827</b>	<b>0.0862</b>	<b>0.2448</b>	<b>0.1673</b>	<b>0.3441</b>	<b>0.1993</b>	0.0413	0.1411	0.0916	0.2227	0.1178
P5-B (5-4)	0.0299	0.1026	0.0665	0.1708	0.0883	0.0506	0.1557	0.1033	0.2350	0.1287	0.0435	0.1316	0.0882	0.2000	0.1102
P5-S (5-5)	0.0574	0.1503	0.1050	0.2207	0.1276	0.0601	0.1611	0.1117	0.2370	0.1360	<u>0.0440</u>	0.1282	0.0865	0.2011	0.1098
P5-B (5-5)	0.0641	0.1794	0.1229	0.2598	0.1488	0.0588	0.1573	0.1089	0.2325	0.1330	0.0386	0.1122	0.0756	0.1807	0.0975
P5-S (5-8)	0.0567	0.1514	0.1049	0.2196	0.1269	0.0571	0.1566	0.1078	0.2317	0.1318	<b>0.0451</b>	0.1322	0.0889	0.2023	0.1114
P5-B (5-8)	<b>0.0726</b>	0.1955	0.1355	0.2802	0.1627	<u>0.0608</u>	0.1564	0.1096	0.2300	0.1332	0.0389	0.1147	0.0767	0.1863	0.0997

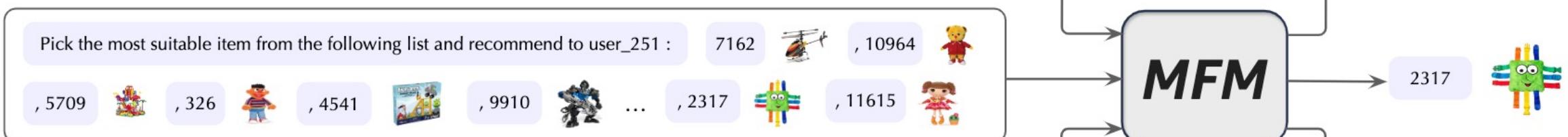
# Easy Handling Multi-modality Information

- Item images can be directly inserted into prompts

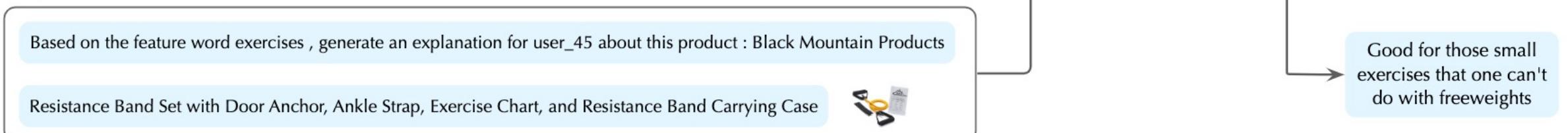
## Sequential Recommendation



## Direct Recommendation

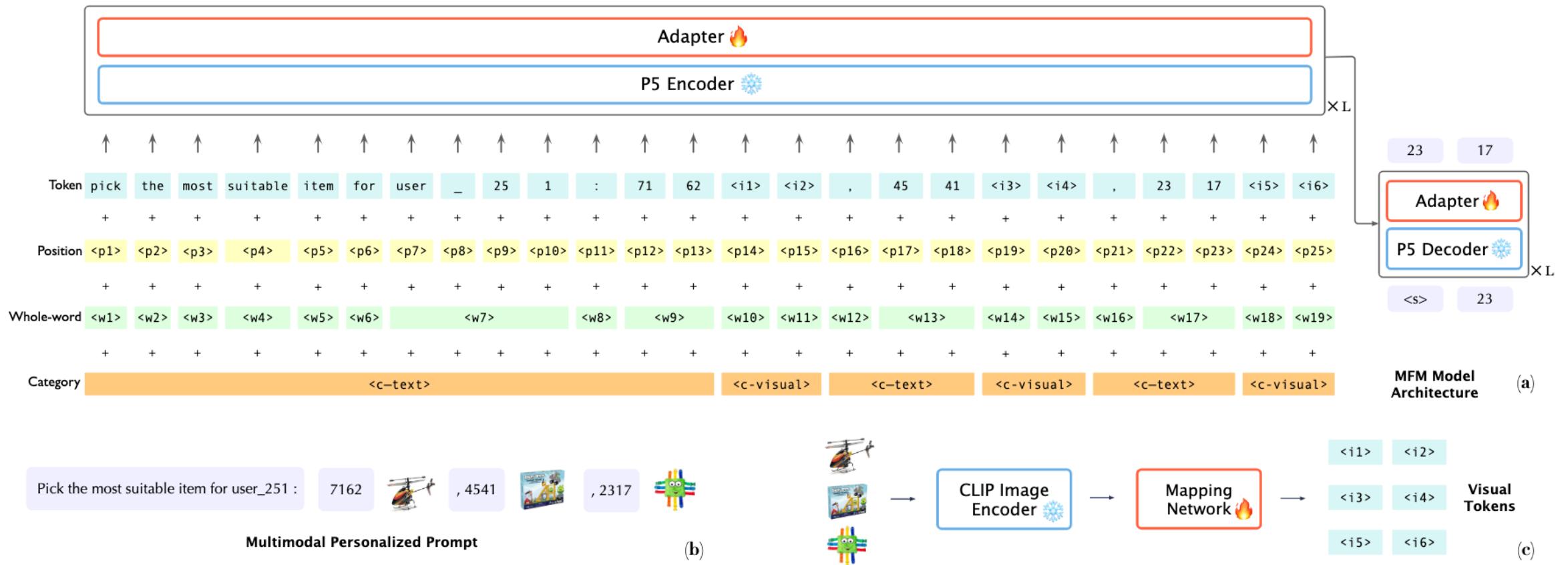


## Explanation Generation



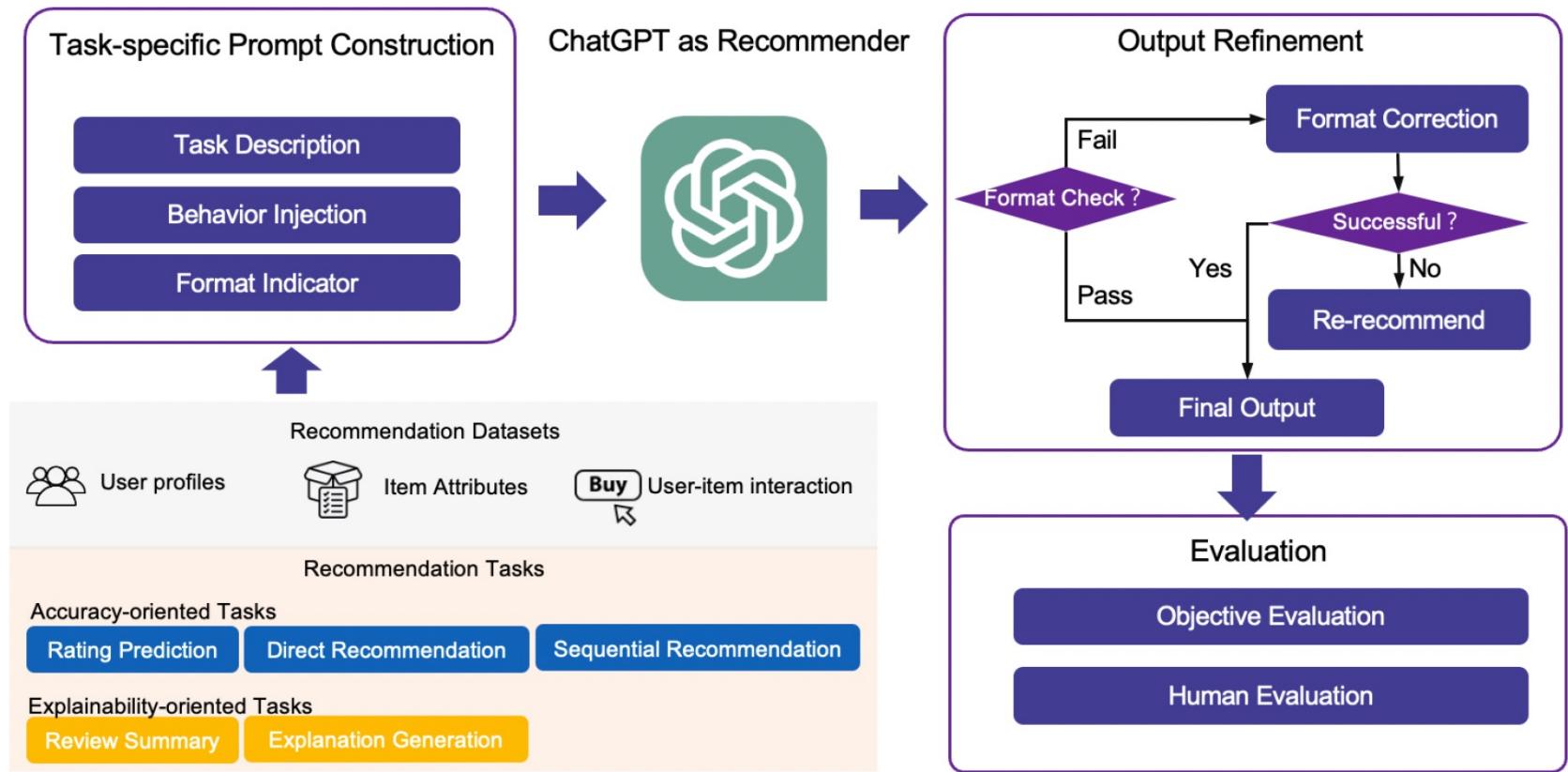
# Easy Handling Multi-modality Information

- Item images can be converted into visual tokens



# ChatGPT as Recommender

- Instruct ChatGPT to perform different tasks w/o fine-tuning
  - Few-shot or zero-shot settings (w/ or w/o demonstration examples)



# ChatGPT on Recommendation Tasks

- Recommendation performance is relatively weak

Sequential Recommendation

Methods	Beauty			
	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.0205	0.0131	0.0347	0.0176
HGN	0.0325	0.0206	0.0512	0.0266
GRU4Rec	0.0164	0.0099	0.0283	0.0137
BERT4Rec	0.0203	0.0124	0.0347	0.0170
FDSA	0.0267	0.0163	0.0407	0.0208
SASRec	0.0387	0.0249	0.0605	0.0318
S <sup>3</sup> -Rec	0.0387	0.0244	0.0647	0.0327
P5-B	<b>0.0493</b>	<b>0.0367</b>	<b>0.0645</b>	<b>0.0416</b>
ChatGPT(zero-shot)	0.0000	0.0000	0.0000	0.0000
ChatGPT(few-shot)	0.0135	0.0135	0.0135	0.0135

Direct Recommendation

Methods	Beauty			
	HR@5	NDCG@5	HR@10	NDCG@10
BPR-MF	0.1426	0.0857	0.2573	0.1224
BPR-MLP	0.1392	0.0848	0.2542	0.1215
SimpleX	<b>0.2247</b>	<b>0.1441</b>	<b>0.3090</b>	<b>0.1711</b>
P5-B	0.1564	0.1096	0.2300	0.1332
ChatGPT(zero-shot)	0.0217	0.0111	0.0652	0.0252
ChatGPT(few-shot)	0.0349	0.0216	0.0930	0.0398

Rating Prediction

Methods	Beauty	
	RMSE	MAE
MF	1.1973	0.9461
MLP	1.3078	0.9597
ChatGPT(zero-shot)	1.4059	1.1861
ChatGPT(few-shot)	<b>1.0751</b>	<b>0.6977</b>

# ChatGPT on Generation Tasks

- Performance with automatic metrics is bad
- Rated highly by human evaluators
  - Existing metrics (BLEU and ROUGE) overly stress the matching between generation and ground-truth [2]

**Explanation Generation**

Methods	Beauty			
	BLUE4	ROUGE1	ROUGE2	ROUGEL
Attn2Seq	0.7889	12.6590	1.6820	9.7481
NRT	0.8295	12.7815	1.8543	9.9477
PETER	1.1541	14.8497	2.1413	11.4143
P5-B	0.9742	16.4530	1.8858	11.8765
PETER+	<b>3.2606</b>	<b>25.5541</b>	<b>5.9668</b>	<b>19.7168</b>
ChatGPT(zero-shot)	0.0000	8.5992	0.6995	4.7564
ChatGPT(few-shot)	1.1967	11.4103	2.5675	5.9119

**Review Summarization**

Methods	Beauty			
	BLUE4	ROUGE1	ROUGE2	ROUGEL
T0	1.2871	1.2750	0.3904	0.9592
GPT-2	0.5879	3.3844	0.6756	1.3956
P5-B	<b>2.1225</b>	<b>8.4205</b>	<b>1.6676</b>	<b>7.5476</b>
ChatGPT(zero-shot)	0.0000	3.8246	0.2857	3.1344
ChatGPT(few-shot)	0.0000	2.7822	0.0000	2.4328

Methods	Evaluators				
	Eva_1	Eva_2	Eva_3	Eva_4	avg_top1 ration
Ground truth	25.0%	45.0%	45.0%	50.0%	38.0%
P5	0.0%	0.0%	0.0%	0.0%	0.0%
ChatGPT(zero-shot)	75.0%	55.0%	55.0%	50.0%	<b>62.0%</b>
					1.46

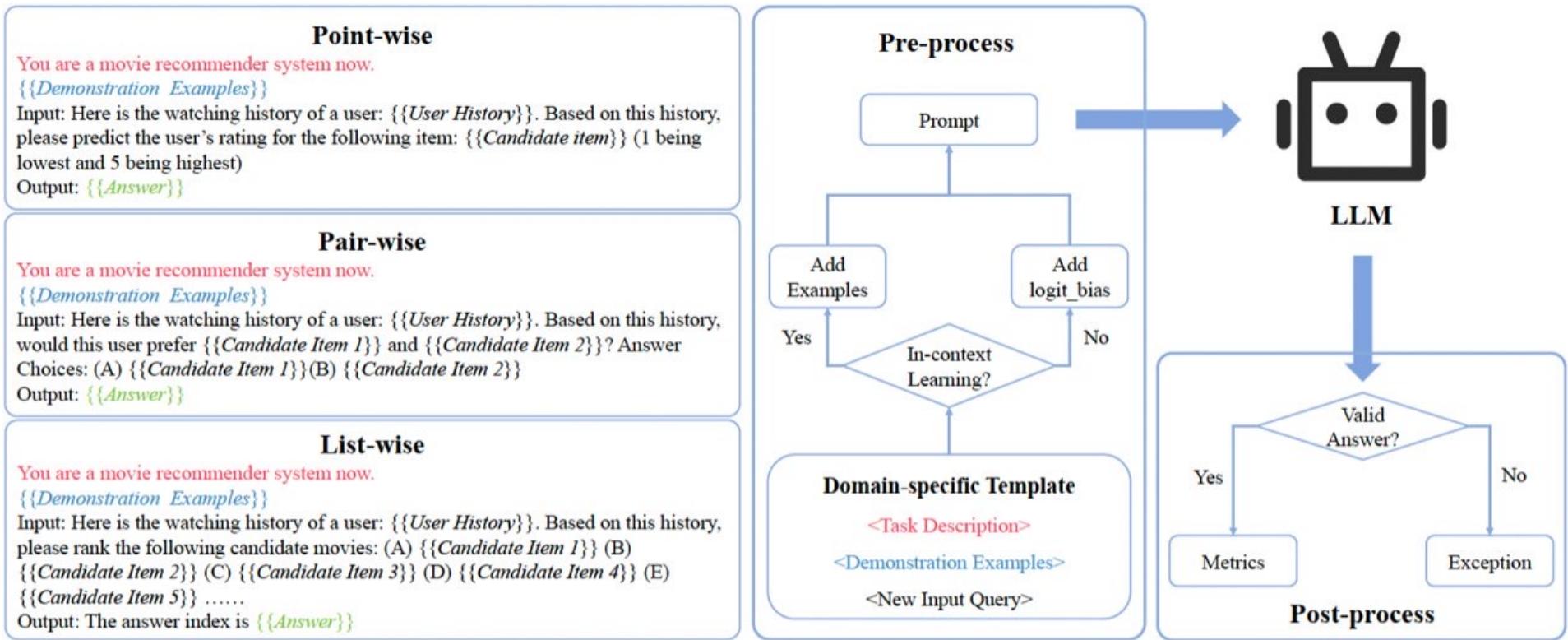
Methods	Evaluators					avg_top1 ration	avg_position
	Eva_1	Eva_2	Eva_3	Eva_4	Eva_5		
Ground truth	12.5%	10.6%	8.7%	17.3%	22.1%	14.2%	2.91
P5	5.8%	0.0%	5.7%	11.5%	19.2%	8.5%	3.16
ChatGPT(zero-shot)	46.2%	37.5%	36.5%	45.2%	23.1%	37.7%	<b>1.90</b>
ChatGPT(few-shot)	35.6%	51.9%	49.0%	26.0%	35.6%	<b>39.6%</b>	2.01

[1] Liu, Junling, et al. "Is chatgpt a good recommender? a preliminary study." *arXiv preprint arXiv:2304.10149* (2023).

[2] Wang, Xiaolei, et al. "Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models." *arXiv preprint arXiv:2305.13112* (2023).

# ChatGPT as Recommender

- Ask ChatGPT to perform three types of recommendation w/o fine-tuning
  - Point-wise (rate), pair-wise (compare), list-wise (rank)



# Recommendation Performance of ChatGPT

- Outperform weak baselines on the three recommendation tasks
  - Random, pop

Domain	Metric	random	pop	text-davinci-002			text-davinci-003			gpt-3.5-turbo (ChatGPT)		
				point-wise	pair-wise	list-wise	point-wise	pair-wise	list-wise	point-wise	pair-wise	list-wise
Movie	Compliance Rate	-	-	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	100.00%
	NDCG@1	0.2000	0.2240	0.3110	0.3203	0.2600	0.2259	0.2843	0.3260	<b>0.3342</b>	0.3230	0.3320
	NDCG@3	0.4262	0.4761	0.5416	0.5728	0.4990	0.4618	0.5441	0.5564	<b>0.5912</b>	0.5827	0.5785
	MRR@3	0.3667	0.4103	0.4824	0.5071	0.4363	0.3998	0.4763	0.4950	<b>0.5260</b>	0.5162	0.5167
Book	Compliance Rate	-	-	99.96%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	99.80%
	NDCG@1	0.2000	0.2440	0.2420	0.2847	0.2000	0.2325	0.2887	0.2440	0.2823	0.3061	<b>0.3126</b>
	NDCG@3	0.4262	0.4999	0.4889	0.5298	0.4290	0.4585	0.5293	0.4597	0.5075	0.5350	<b>0.5395</b>
	MRR@3	0.3667	0.4340	0.4247	0.4646	0.3690	0.3993	0.4665	0.4040	0.4495	0.4774	<b>0.4800</b>
Music	Compliance Rate	-	-	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.80%
	NDCG@1	0.2000	0.1780	0.2354	0.2397	0.2300	0.2377	0.2690	0.2540	0.2892	0.3077	<b>0.3086</b>
	NDCG@3	0.4262	0.4094	0.4623	0.4681	0.4277	0.4732	0.5072	0.4506	0.5201	0.5439	<b>0.5567</b>
	MRR@3	0.3667	0.3470	0.4030	0.4082	0.3750	0.4113	0.4448	0.4000	0.4605	0.4830	<b>0.4950</b>
News	Compliance Rate	-	-	99.80%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.60%
	NDCG@1	0.2000	<b>0.3080</b>	0.2183	0.2200	0.2920	0.2532	0.2630	0.2540	0.2591	0.2491	0.2892
	NDCG@3	0.4262	<b>0.5444</b>	0.4483	0.4550	0.5059	0.4880	0.4892	0.4742	0.4826	0.4991	<b>0.5094</b>
	MRR@3	0.3667	<b>0.4840</b>	0.3879	0.3936	0.4497	0.4271	0.4294	0.4173	0.4246	0.4354	<b>0.4515</b>

# With Fine-tuning or Without Fine-tuning

- Without fine-tuning, LLM cannot easily solve RS problems
  - RS is a highly specialized area that requires collaborative knowledge, which LLM did not learn during the pre-training stage [1]
- RS practitioners do **not** have an existential crisis as NLP community
  - Many NLP problems can be easily addressed by LLM
  - RS is still an open problem and will evolve with LLM

# Role of LLM in Recommendation

- LLM as RS
  - E.g., P5 and ChatGPT-based recommenders
- LLM in RS as a component

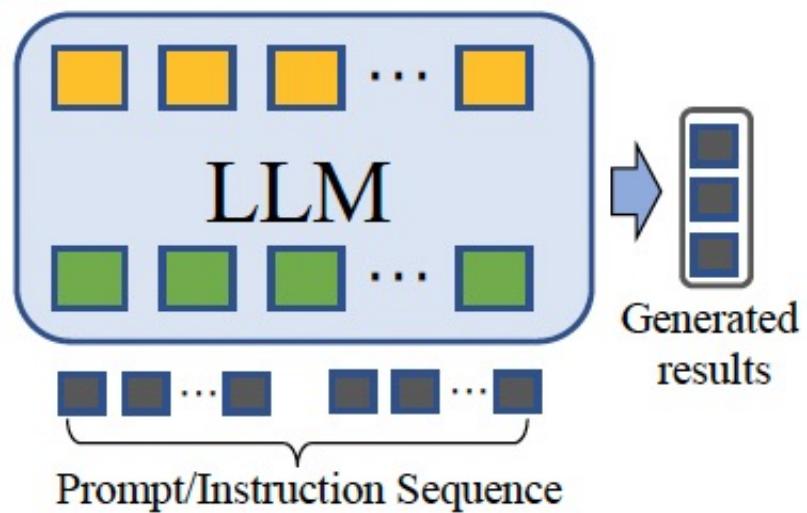


Image credit to [1]

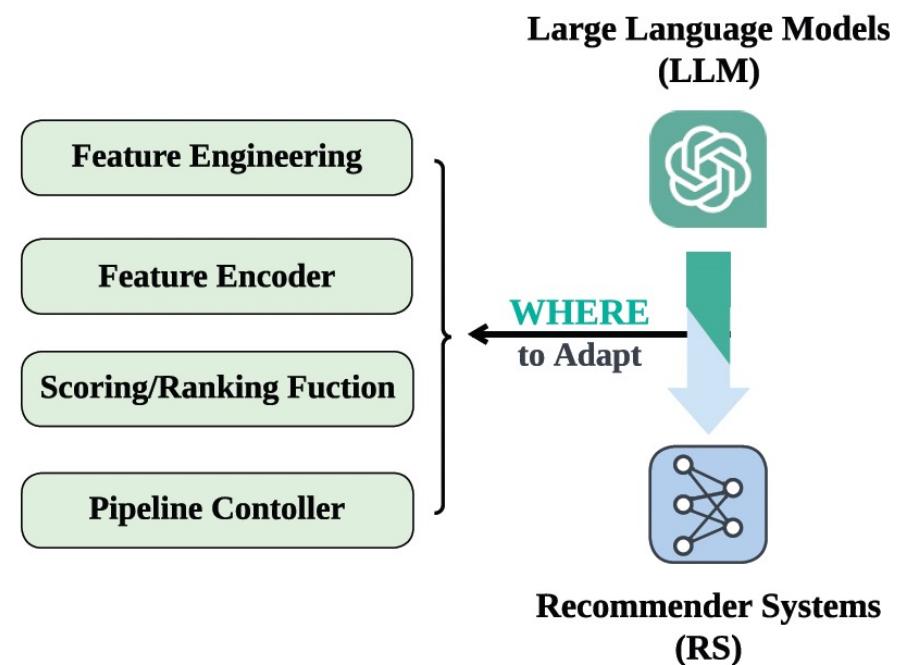


Image credit to [2]

# LLM as Feature Encoder

- LLM is grounded to recommendation space by generating tokens for items
- Then these tokens are grounded to actual items in the actual item space

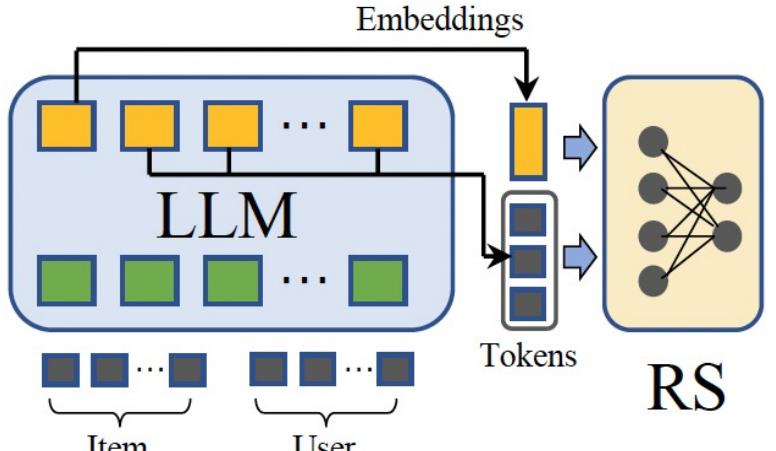


Image credit to [2]



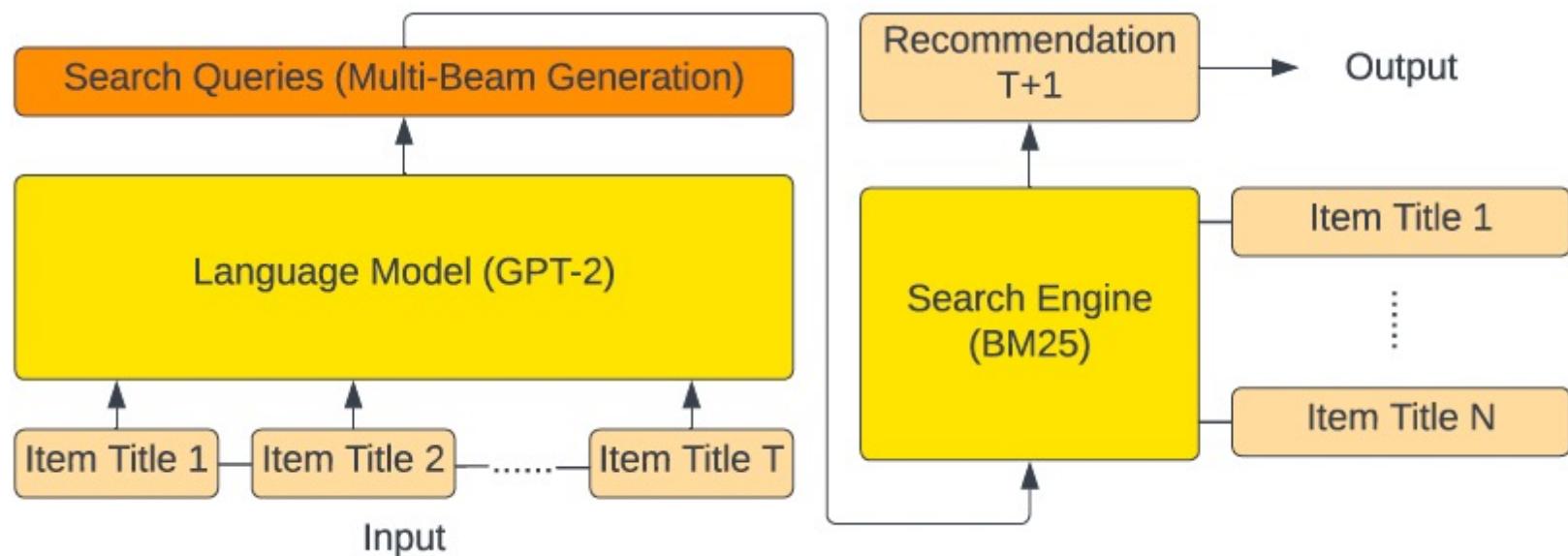
Image credit to [1]

[1] Bao, Keqin, et al. "A bi-step grounding paradigm for large language models in recommendation systems." *arXiv preprint arXiv:2308.08434* (2023).

[2] Wu, Likang, et al. "A Survey on Large Language Models for Recommendation." *arXiv preprint arXiv:2305.19860* (2023).

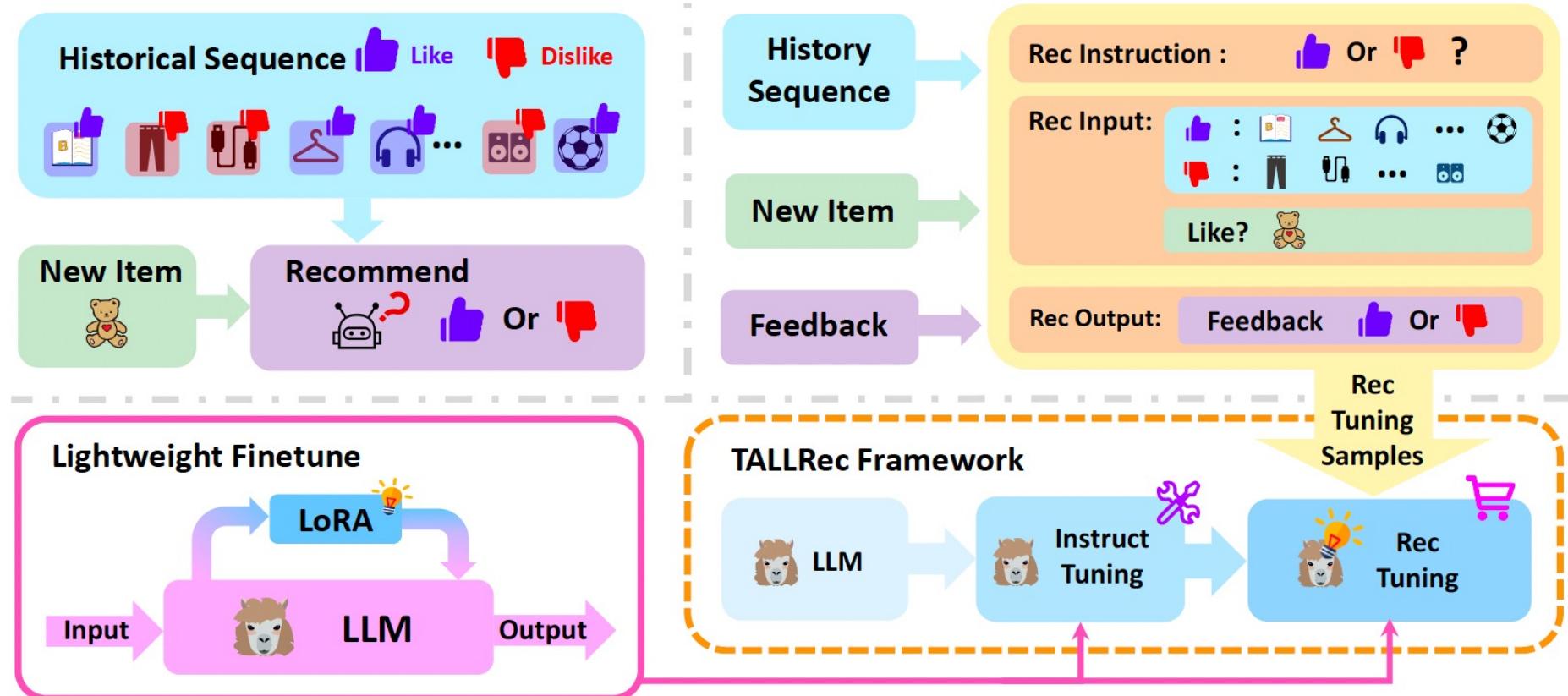
# LLM as Feature Encoder

- Instruct LLM to generate search queries
- Then a searching algorithm is applied to retrieve items based on the queries



# LLM as Scoring Function

- Instruct LLM to generate a binary score (like or dislike) for each item
  - Discriminative as traditional recommenders



# LLM as Ranking Function

- Provide LLM with candidates from another RS for re-ranking

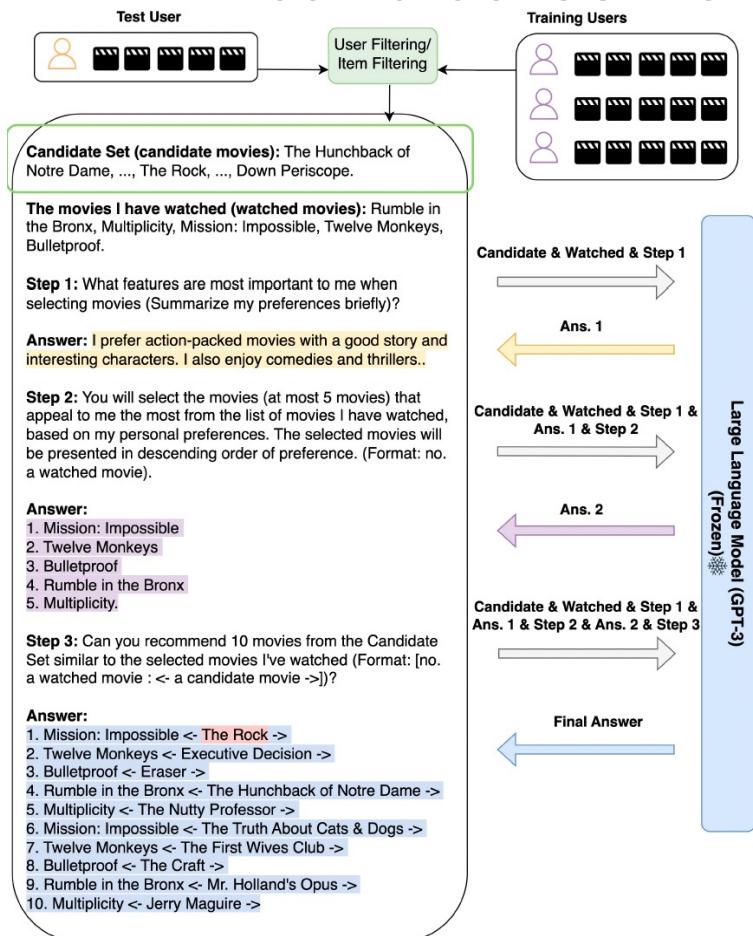


Image credit to NIR [1]

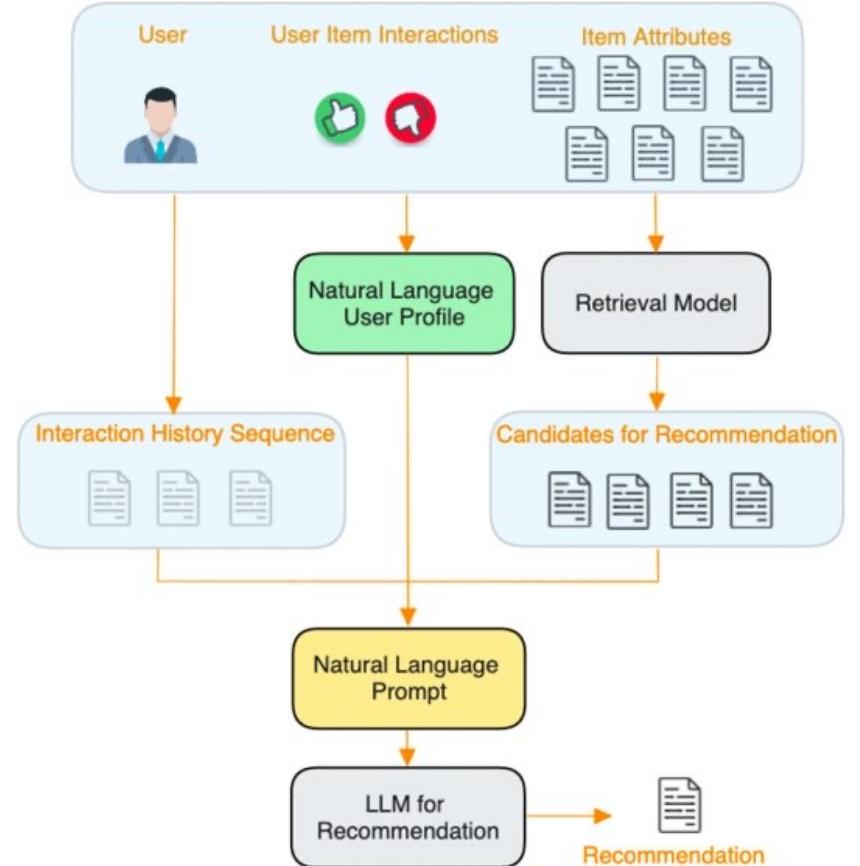


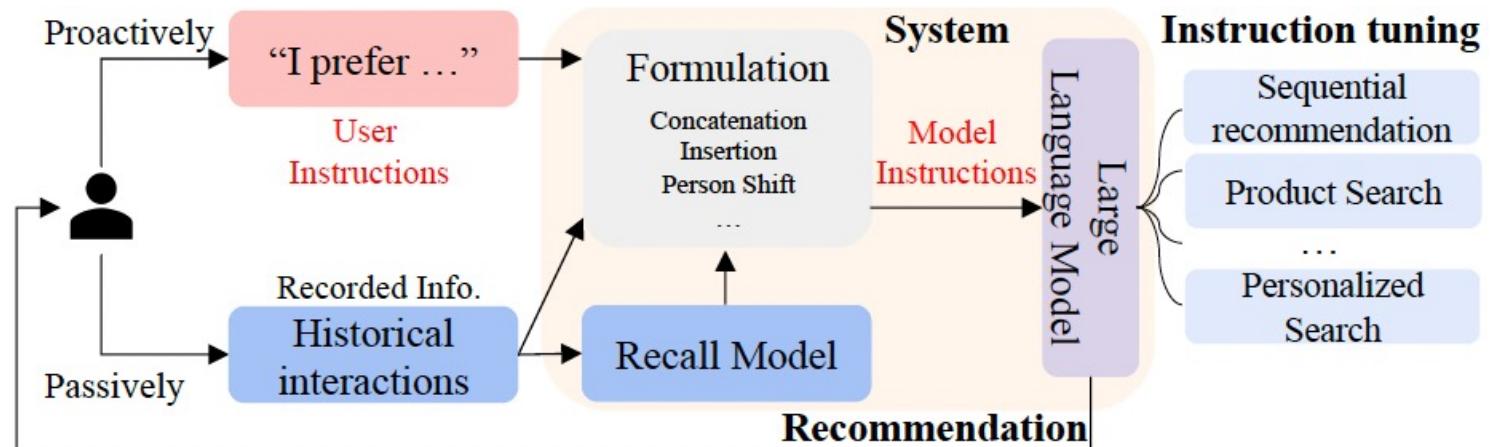
Image credit to PALR [2]

[1] Wang, Lei, and Ee-Peng Lim. "Zero-Shot Next-Item Recommendation using Large Pretrained Language Models." *arXiv preprint arXiv:2304.03153* (2023).

[2] Chen, Zheng. "PALR: Personalization Aware LLMs for Recommendation." Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval (2023).

# LLM as Ranking Function

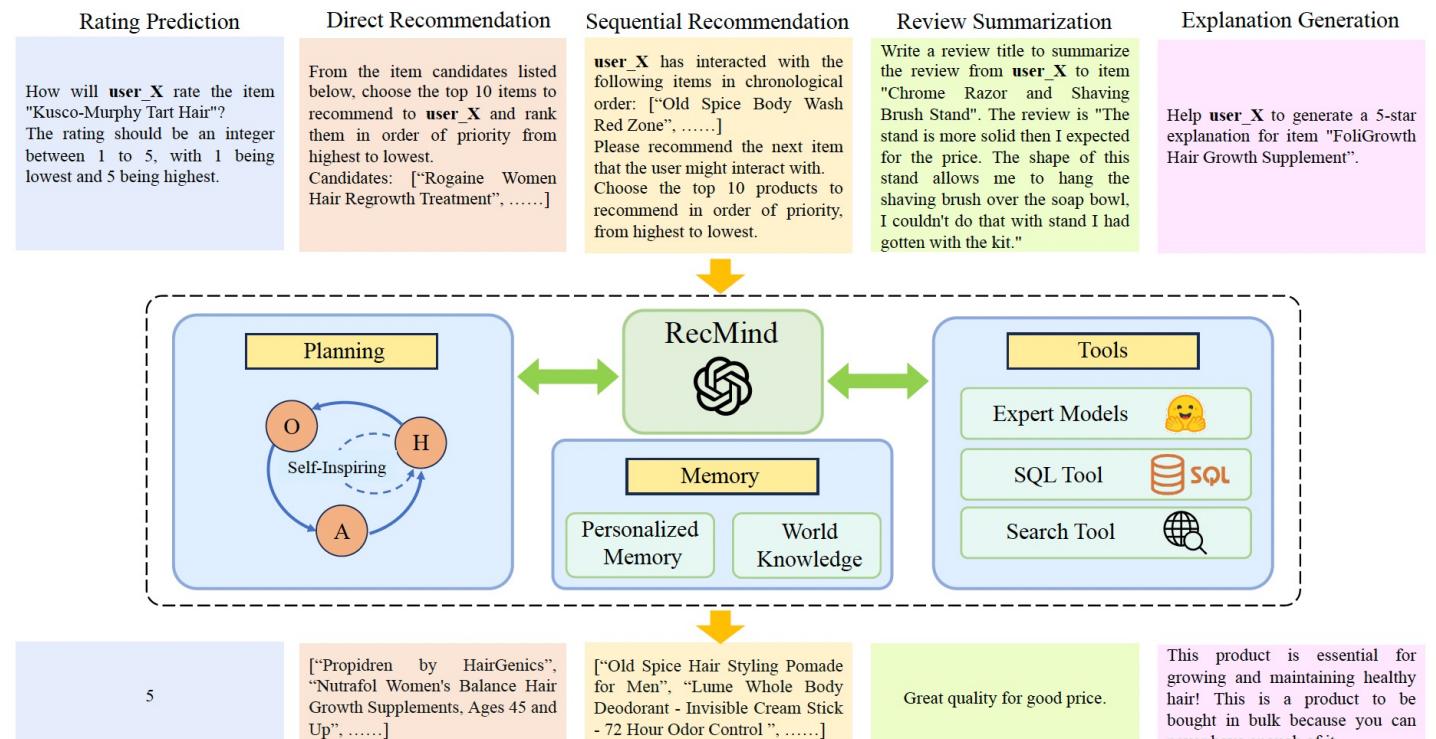
- LLM takes candidates from a Recall model for re-ranking
  - Design prompts for different recommendation settings



Instantiation	Model Instructions
$\langle P_1, I_0, T_0 \rangle$	The user has purchased these items: <historical interactions>. Based on this information, is it likely that the user will interact with <target item> next?
$\langle P_2, I_0, T_3 \rangle$	You are a search engine and you meet a user's query: <explicit preference>. Please respond to this user by selecting items from the candidates: <candidate items>.
$\langle P_0, I_1, T_2 \rangle$	As a recommender system, your task is to recommend an item that is related to the user's <vague intention>. Please provide your recommendation.
$\langle P_0, I_2, T_2 \rangle$	Suppose you are a search engine, now the user search that <specific Intention>, can you generate the item to respond to user's query?
$\langle P_1, P_2, T_2 \rangle$	Here is the historical interactions of a user: <historical interactions>. His preferences are as follows: <explicit preference>. Please provide recommendations .
$\langle P_1, I_1, T_2 \rangle$	The user has interacted with the following <historical interactions>. Now the user search for <vague intention>, please generate products that match his intent.
$\langle P_1, I_2, T_2 \rangle$	The user has recently purchased the following <historical items>. The user has expressed a desire for <specific intention>. Please provide recommendations.

# LLM as Pipeline Controller

- Break each task into several planning steps
  - Thought, action and observation
- Control personalized memory and world knowledge
- Perform specific tasks with tools, e.g., task-specific models



# Recommendation Tasks

- Rating Prediction
- Sequential Recommendation
- Top-N Recommendation
- Explanation Generation
- Review Summarization
- Review Generation
- **Conversational Recommendation**

# Conversational Recommendation

- LLM as the whole conversational recommender
  - T: Task description
  - F: Format requirement
  - S: Conversational context

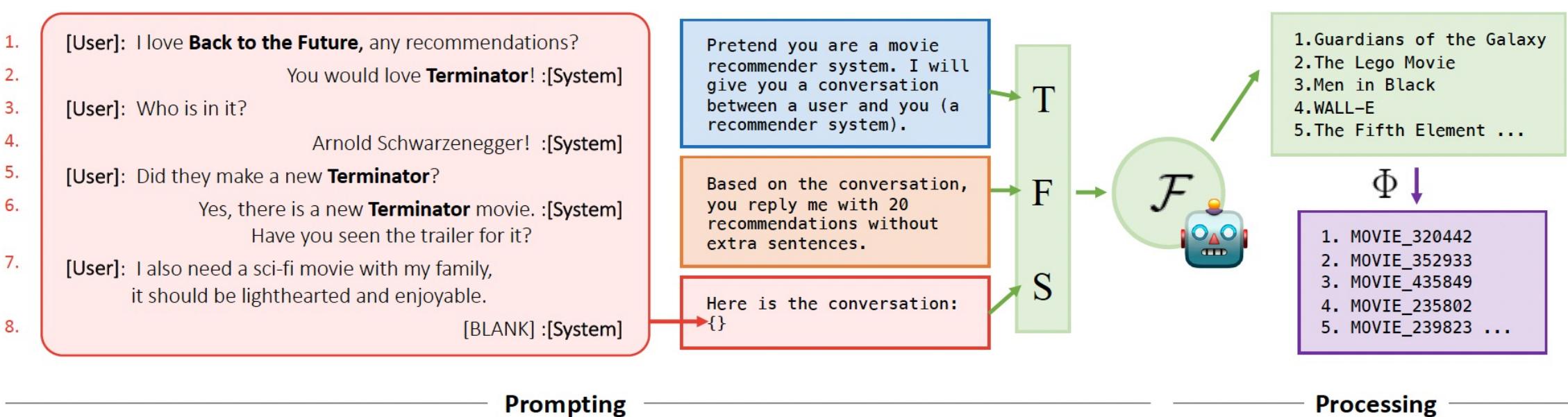


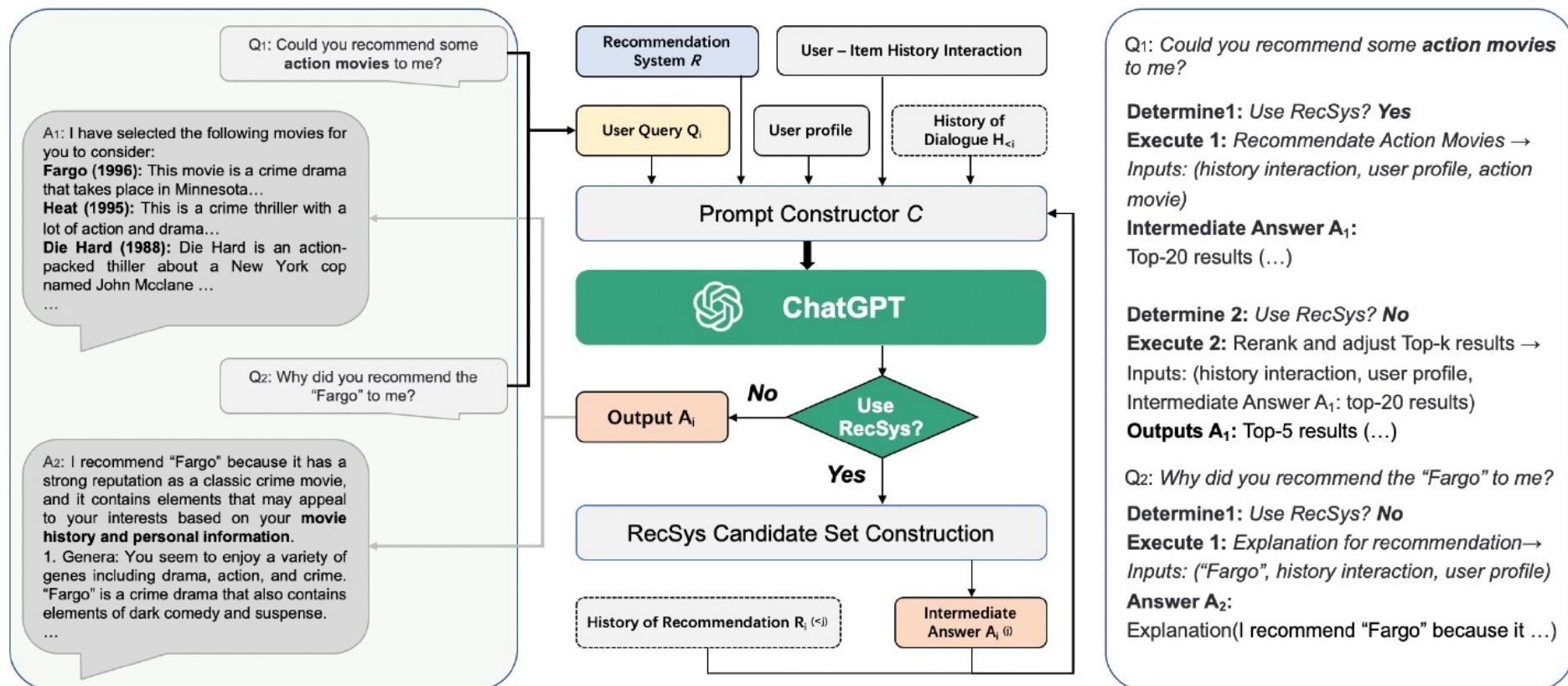
Image credit to [1]

[1] He, Zhankui, et al. "Large Language Models as Zero-Shot Conversational Recommenders." CIKM 2023.

[2] Cui, Zeyu, et al. "M6-rec: Generative pretrained language models are open-ended recommender systems." *arXiv preprint arXiv:2205.08084* (2022).

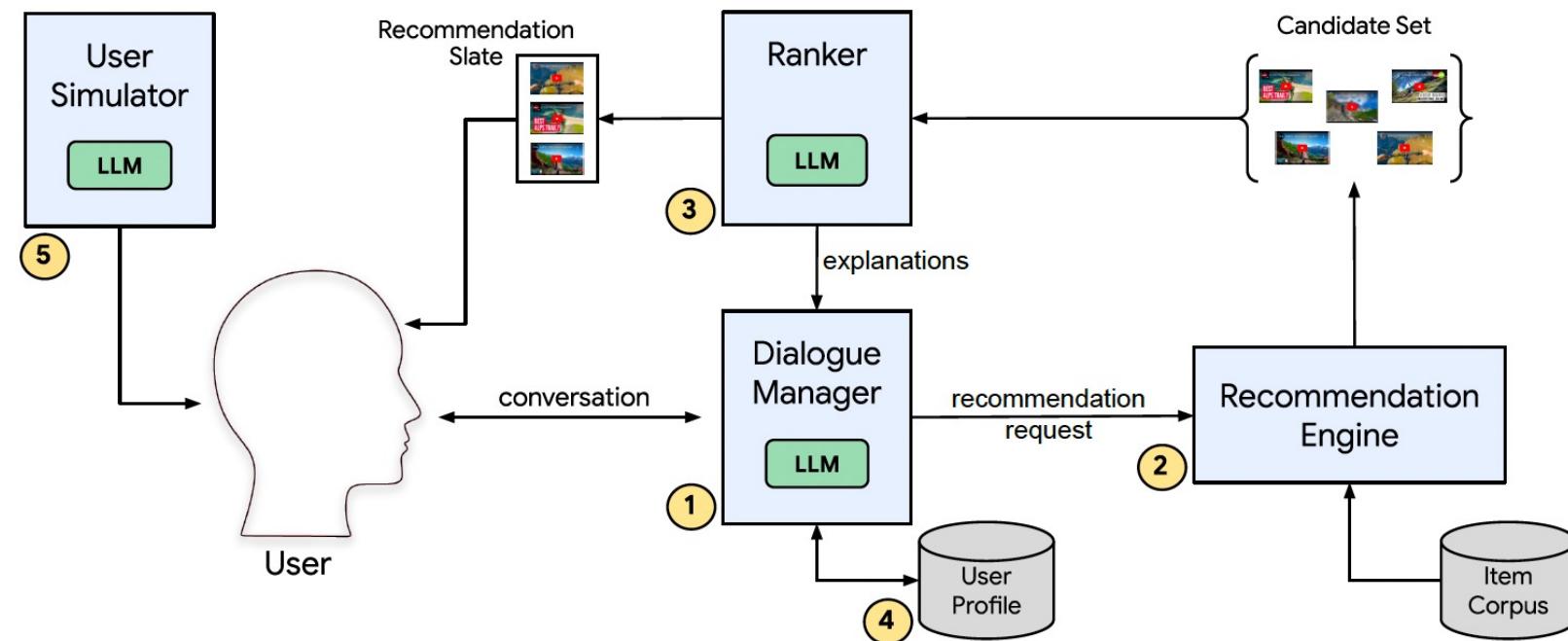
# Conversational Recommendation

- LLM as dialogue manager that merges various types of info
  - Recommendations (from another model)
  - Dialogue history

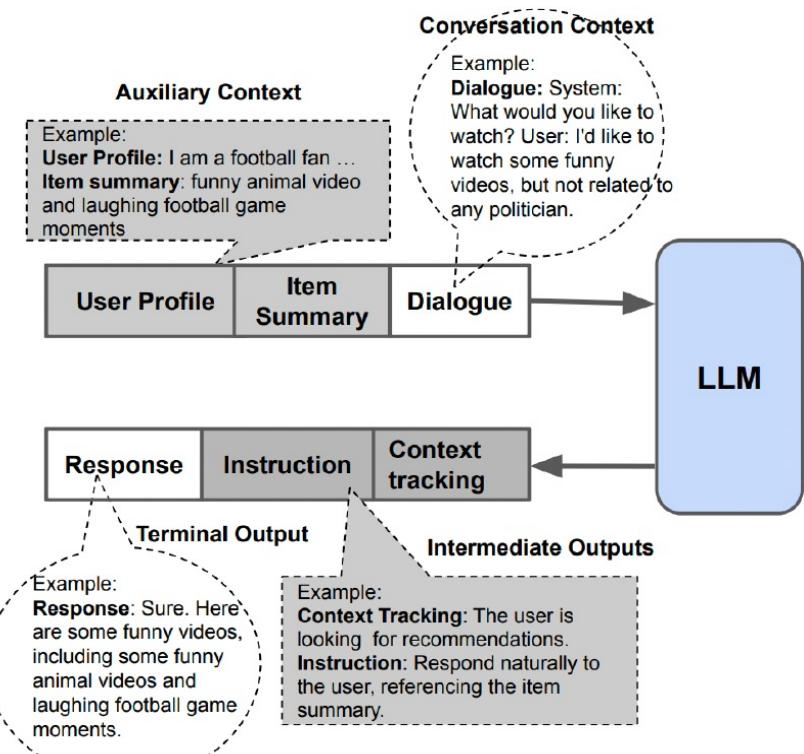


# Conversational Recommendation

- Multiple LLMs play separate roles
  - Dialogue Manager
  - Ranking Function
  - User Simulator



## Dialogue Manager



# Evaluation Protocols

- Recommendation
  - RMSE and MAE for rating prediction
  - NDCG, Precision and Recall for top-N and sequential recommendation
  - Online A/B test
- Generation
  - BLEU and ROUGE for text similarity
    - Overly stress the matching between generation and ground-truth [1]
    - Advanced metrics are needed
  - Human Evaluation

# Trustworthy LLMs for Recommendation

# Trustworthy LLM4RS

- Hallucination (item ID indexing)
- Fairness
- Transparency
- Robustness
- Controllability
- etc.

## Hallucination



Large Language Model



Spigen Liquid Air Armor  
Designed for OnePlus 8  
Case (2020) [NOT  
Compatible with Verizon  
UW Version] - Matte  
Black  
★★★★★ 673  
INR926.48



Spigen Ultra Hybrid  
Designed for OnePlus 8  
Case (2020) [NOT  
Compatible with Verizon  
UW Version] - Crystal  
Clear  
★★★★★ 1,406  
INR926.48



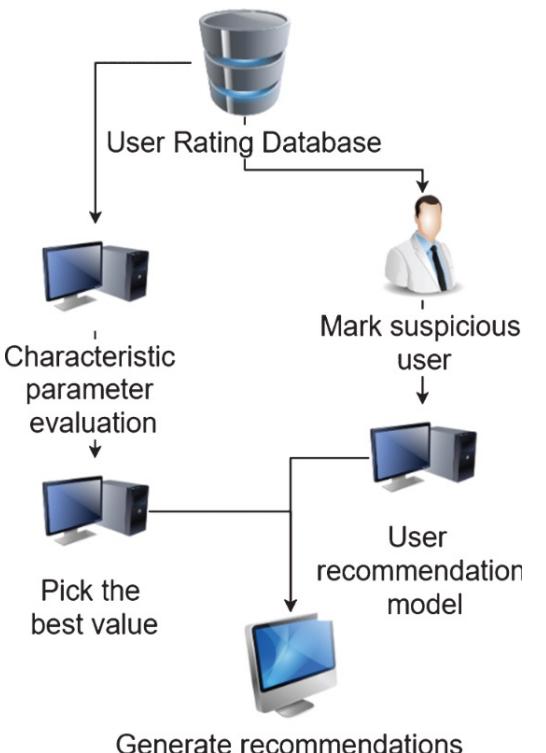
SAMSUNG Galaxy A71  
5G Factory Unlocked  
Android Cell Phone  
128GB US Version  
Smartphone 6.7-inch  
AMOLED Screen Long...  
★★★★★ 2,021  
INR45,048.27

## Fairness

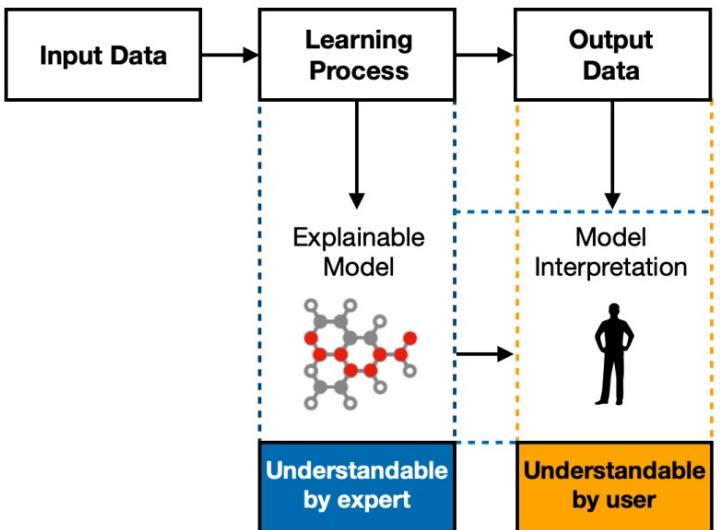
Can I be treated fairly on age?



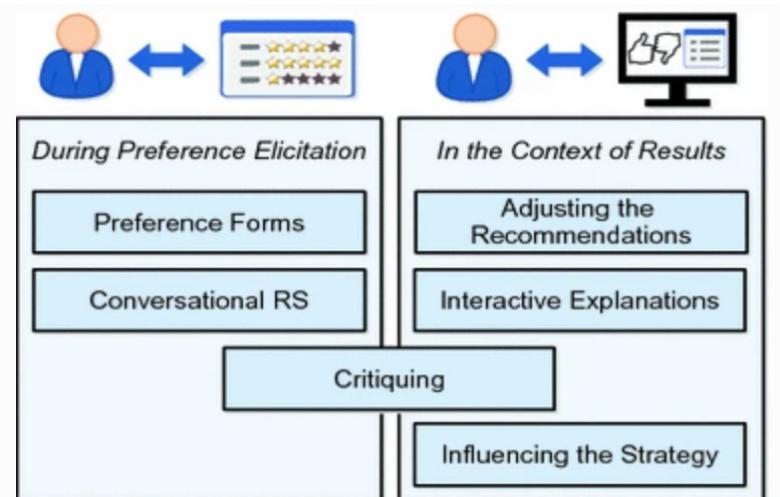
## Robustness



## Transparency



## Controllability



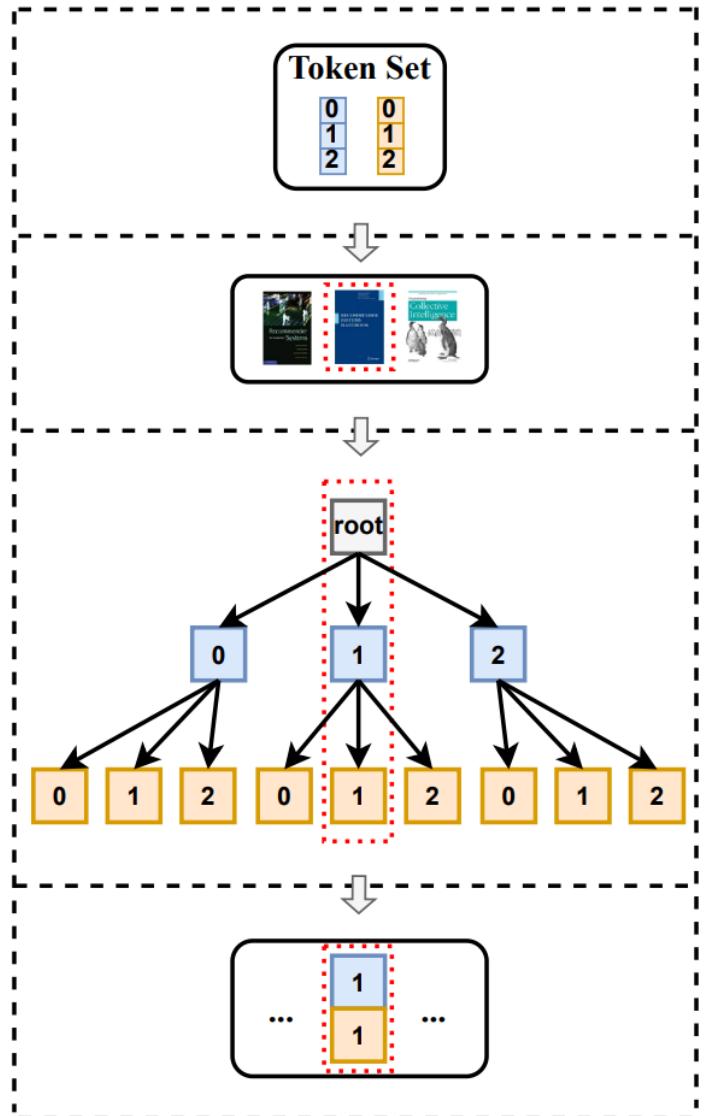
# Hallucination: Item Generation

- LLM-based Generative Recommendation Paradigm
  - We want to **directly generate** the recommended item
  - Avoid one-by-one ranking score calculation
- However, item descriptions can be very long
  - e.g., product description: >100 words
  - e.g., news article: >1,000 words

# Hallucination: Item Generation

- Generating long text is difficult, especially for recommendation
  - Hallucination problem
  - Generated text does not correspond to a real existing item in database
  - Calculating similarity between generated text and item text?
  - Goes back to one-by-one similarity calculation for ranking!
- Item ID: A short sequence of tokens for an item
  - Easy generation, and can be indexed!
- Item ID can take various forms
  - A sequence of numerical tokens <73><91><26>
  - A sequence of word tokens <the><lord><of><the><rings>

# Why Item IDs can eliminate hallucination?

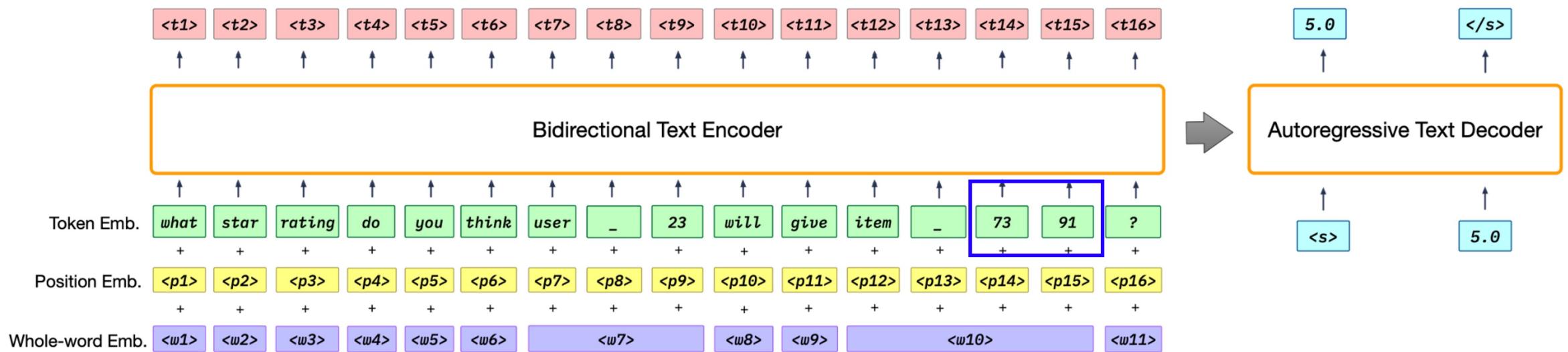


With item indices consisting of a limited vocabulary and known structure, we can **constrain the beam search** over limited allowed tokens for every generation step.

Thus, hallucination will be eliminated.

# How to Index Items?

- Item ID: item needs to be represented as a sequence of tokens
  - e.g., an item represented by two tokens `<73>` `<91>`



- Different item indexing gives very different performance

# How to Index Items (create Item IDs)

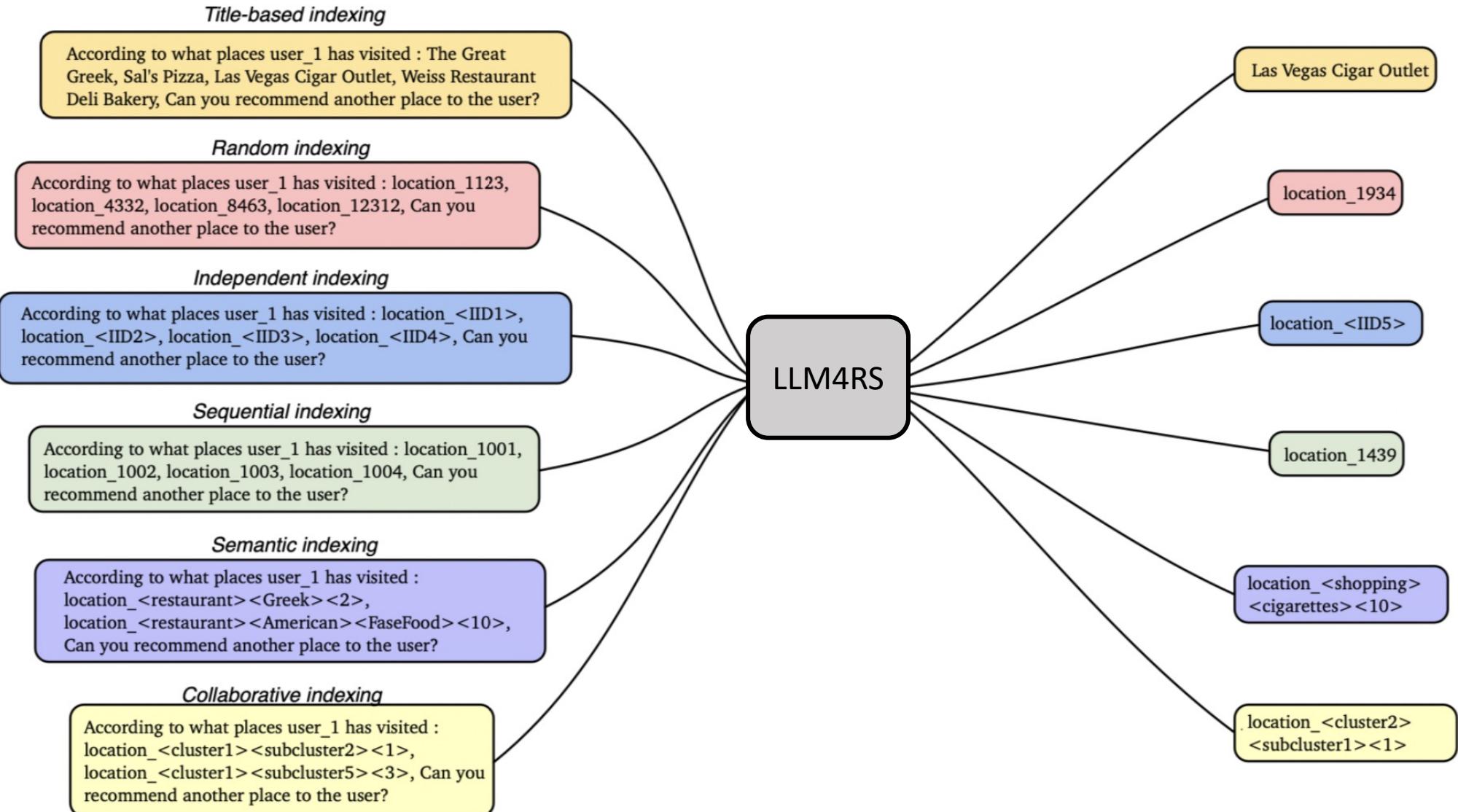
- Three properties for good item indexing methods
  - Items are distinguishable (different items have different IDs)
  - Similar items have similar IDs (more shared tokens in their IDs)
  - Dissimilar items have dissimilar IDs (less shared tokens in their IDs)
- Three naïve Indexing methods
  - Random ID (RID): Item <73><91>, item <73><12>, ...
  - Title as ID (TID): Item <the><lord><of><the><rings>, ...
  - Independent ID (IID): Item <1364>, Item <6321>, ...

Method	Amazon Sports				Amazon Beauty				Yelp			
	HR@5	NCDG@5	HR@10	NCDG@10	HR@5	NCDG@5	HR@10	NCDG@10	HR@5	NCDG@5	HR@10	NCDG@10
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0170	0.0110	0.0284	0.0147
S <sup>3</sup> -Rec	0.0251	<b>0.0161</b>	0.0385	<b>0.0204</b>	0.0387	0.0244	<b>0.0647</b>	0.0327	0.0201	0.0123	0.0341	0.0168
RID	0.0208	0.0122	0.0288	0.0153	0.0213	0.0178	0.0479	0.0277	0.0225	<b>0.0159</b>	0.0329	0.0193
TID	0.0000	0.0000	0.0000	0.0000	0.0182	0.0132	0.0432	0.0254	0.0058	0.0040	0.0086	0.0049
IID	<b>0.0268</b>	0.0151	<b>0.0386</b>	0.0195	<b>0.0394</b>	<b>0.0268</b>	0.0615	<b>0.0341</b>	<b>0.0232</b>	0.0146	<b>0.0393</b>	<b>0.0197</b>

# How to Index Items (create Item IDs)

- Three naïve Indexing methods
  - Random ID (RID): Item <73><91>, item <73><29>, ...
    - Very different items may share the same tokens
    - Mistakenly making them semantically similar
  - Title as ID (TID): Item <the><lord><of><the><rings>
    - Very different movies may share similar titles
      - Inside Out (animation) and Inside Job (documentary)
      - The Lord of the Rings (epic fantasy) and The Lord of War (crime drama)
  - Independent ID (IID): Item <1364>, Item <6321>, ...
    - Too many out-of-vocabulary (OOV) new tokens need to learn
    - Computationally unscalable

# Meticulous Item Indexing Methods are Needed



# Sequential Indexing (SID)

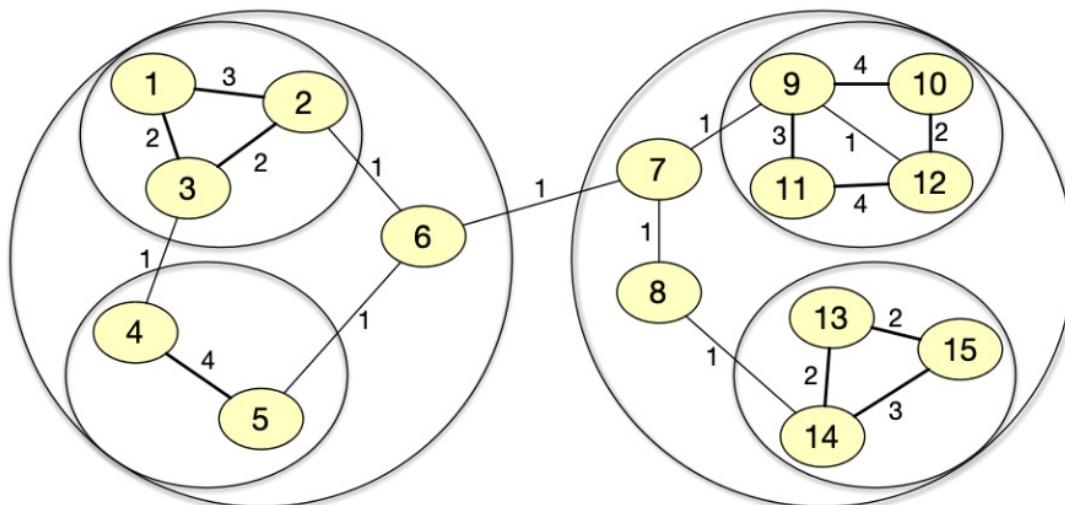
- Leverage the **local** co-appearance information between items

Training Sequence											Validation	Testing
User 1	1001	1002	1003	1004	1005	1006	1007	1008	1009		1018	1019
User 2	1010	1011	1001	1012	1008	1009	1013	1014			1022	1023
User 3	1015	1016	1017	1007	1018	1019	1020	1021	1009		1015	1016
User 4	1022	1023	1005	1002	1006	1024					1002	1008
User 5	1025	1026	1027	1028	1029	1030	1024	1020	1021	1031	1033	1034

- After tokenization, co-appearing items share similar tokens
  - Item 1004: <100><4>
  - Item 1005: <100><5>

# Collaborative Indexing (CID)

- Leverage the **global** co-appearance information between items
  - Spectral Matrix Factorization over the item-item co-appearance matrix
  - Hierarchical Spectral Clustering



(a) Recursive spectral clustering on item co-appearance graph

(b) Adjacency matrix

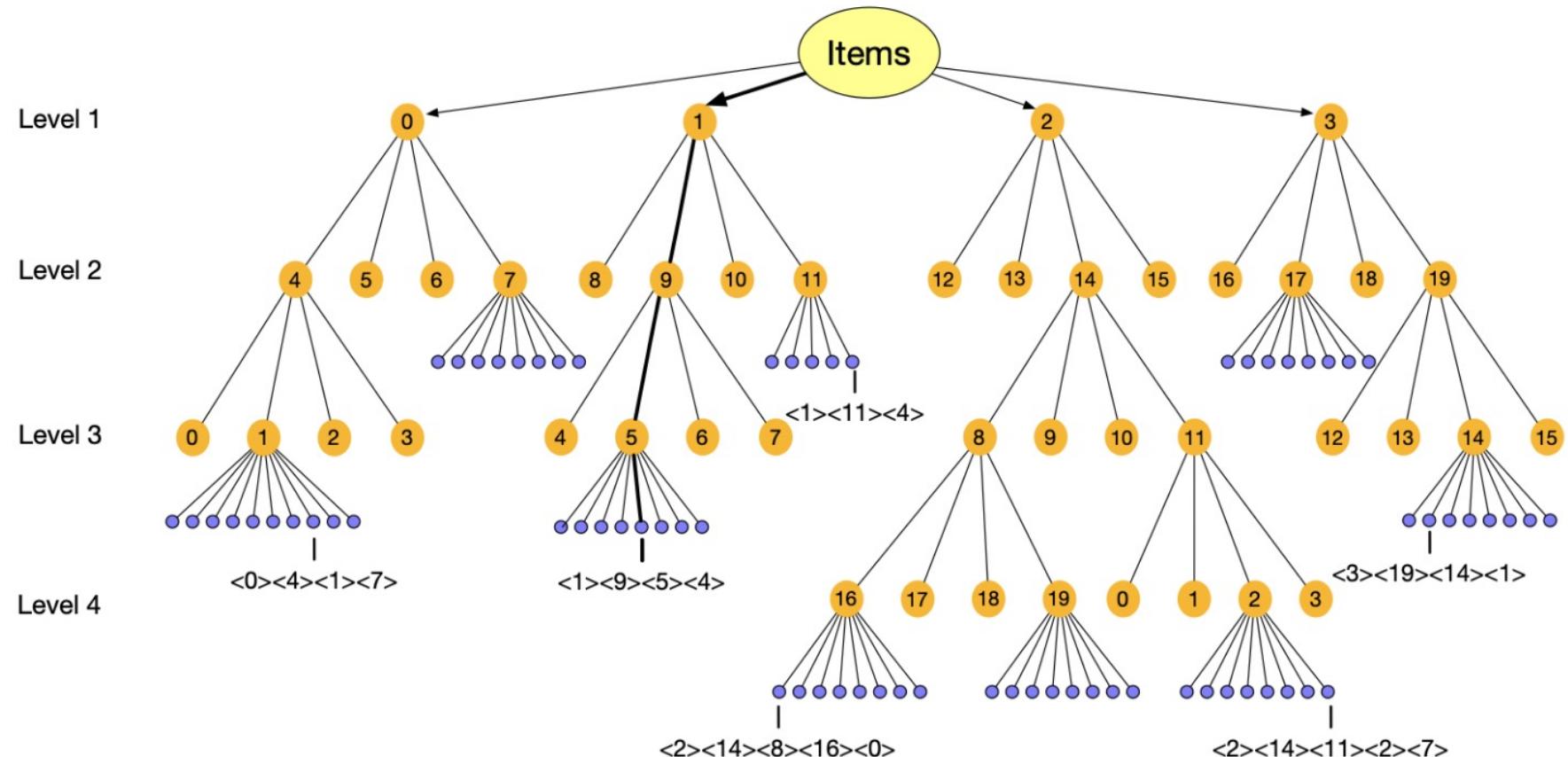
$$\begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & \dots \\ 1 & \left[ \begin{array}{cccccc} 0 & 3 & 2 & 0 & 0 & 0 & \dots \\ 3 & 0 & 2 & 0 & 0 & 1 & \dots \\ 2 & 2 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 4 & 0 & \dots \\ 0 & 0 & 0 & 4 & 0 & 1 & \dots \\ 0 & 1 & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right] \end{matrix}$$

(c) Laplacian matrix

$$\begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & \dots \\ 1 & \left[ \begin{array}{cccccc} 5 & -3 & -2 & 0 & 0 & 0 & \dots \\ -3 & 6 & -2 & 0 & 0 & -1 & \dots \\ -2 & -2 & 6 & -1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 5 & -4 & 0 & \dots \\ 0 & 0 & 0 & -4 & 5 & -1 & \dots \\ 0 & 1 & 0 & 0 & -1 & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \right] \end{matrix}$$

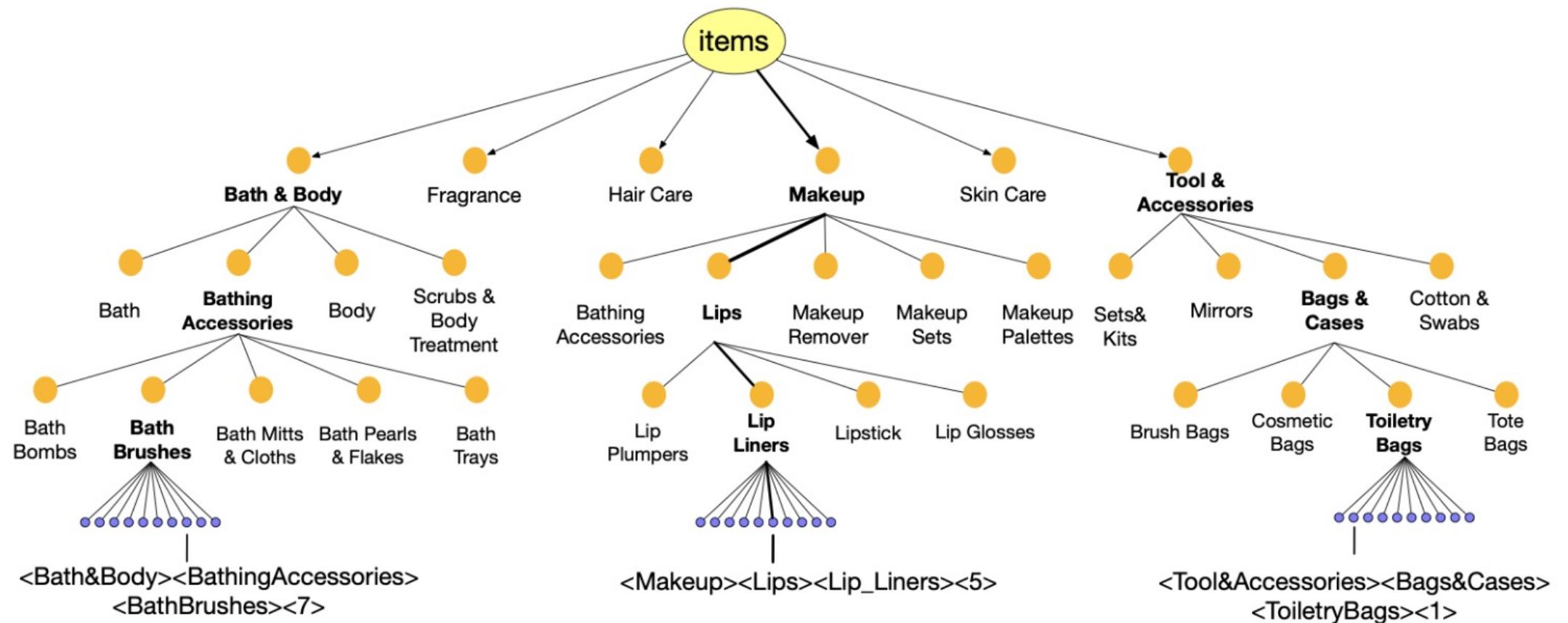
# Collaborative Indexing (CID)

- Leverage the **global co-appearance information between items**
  - Root-to-Leaf Path-based Indexing
  - Items in the same cluster share more tokens



# Semantic (Content-based) Indexing (SemID)

- Leverage the **item content information** for item indexing
  - e.g., the **multi-level item category** information in Amazon



# Hybrid Indexing (HID)

- Concatenate more than one of the following indices
  - Random ID (RID)
  - Title as ID (TID)
  - Independent ID (IID)
  - Sequential ID (SID)
  - Collaborative ID (CID)
  - Semantic ID (SemID)
- For example, if an item's Semantic ID and Collaborative ID are as follows:
  - SemID: <Makeup><Lips><Lip\_Liners><5>
  - CID: <1><9><5><4>
- Then its Hybrid ID is <Makeup><Lips><Lip\_Liners><1><9><5><4>

# Different Item Indexing Gives Different Performance

Method	Amazon Sports				Amazon Beauty				Yelp				
	HR@5 NCDG@5		HR@10 NCDG@10		HR@5 NCDG@5		HR@10 NCDG@10		HR@5 NCDG@5		HR@10 NCDG@10		
Naïve indexing methods	Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.015	0.0099	0.0263	0.0134
	HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0186	0.0115	0.0326	0.159
	GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0176	0.0110	0.0285	0.0145
	BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0051	0.0033	0.0090	0.0090
	FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0158	0.0098	0.0276	0.0136
	SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0170	0.0110	0.0284	0.0147
	$S^3$ -Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0201	0.0123	0.0341	0.0168
Advanced indexing methods	RID	0.0208	0.0122	0.0288	0.0153	0.0213	0.0178	0.0479	0.0277	<u>0.0225</u>	<u>0.0159</u>	<u>0.0329</u>	<u>0.0193</u>
	TID	0.000	0.000	0.000	0.000	0.0182	0.0132	0.0432	0.0254	<u>0.0058</u>	<u>0.0040</u>	<u>0.0086</u>	<u>0.0049</u>
	IID	0.0268	0.0151	<u>0.0386</u>	0.0195	<u>0.0394</u>	<u>0.0268</u>	0.0615	<u>0.0341</u>	0.0232	0.0146	<u>0.0393</u>	<u>0.0197</u>
Hybrid indexing methods	SID	<u>0.0264</u>	<u>0.0186</u>	0.0358	<u>0.0216</u>	<u>0.0430</u>	<u>0.0288</u>	0.0602	<u>0.0368</u>	<b>0.0346</b>	<b>0.0242</b>	<b>0.0486</b>	<b>0.0287</b>
	CID	<u>0.0313</u>	<u>0.0224</u>	<u>0.0431</u>	<u>0.0262</u>	<u>0.0489</u>	<u>0.0318</u>	<u>0.0680</u>	<u>0.0357</u>	<u>0.0261</u>	<u>0.0171</u>	<u>0.0428</u>	<u>0.0225</u>
	SemID	<u>0.0274</u>	<u>0.0193</u>	<u>0.0406</u>	<u>0.0235</u>	<u>0.0433</u>	<u>0.0299</u>	<u>0.0652</u>	<u>0.0370</u>	<u>0.0202</u>	<u>0.0131</u>	<u>0.0324</u>	<u>0.0170</u>
	SID+IID	0.0235	0.0161	0.0339	0.0195	<u>0.0420</u>	<u>0.0297</u>	0.0603	<u>0.0355</u>	<u>0.0329</u>	<u>0.0236</u>	<u>0.0465</u>	<u>0.0280</u>
	CID+IID	<b>0.0321</b>	<b>0.0227</b>	<b>0.0456</b>	<b>0.0270</b>	<b>0.0512</b>	<b>0.0356</b>	<b>0.0732</b>	<b>0.0427</b>	<u>0.0287</u>	<u>0.0195</u>	<u>0.0468</u>	<u>0.0254</u>
	SemID+IID	<u>0.0291</u>	<u>0.0196</u>	<u>0.0436</u>	<u>0.0242</u>	<u>0.0501</u>	<u>0.0344</u>	<u>0.0724</u>	<u>0.0411</u>	<u>0.0229</u>	<u>0.0150</u>	<u>0.0382</u>	<u>0.0199</u>
	SemID+CID	0.0043	0.0031	0.0070	0.0039	0.0355	0.0248	0.0545	0.0310	0.0021	0.0016	0.0056	0.0029

- Advanced indexing methods are better than naïve methods
- Some hybrid indexing can further improve performance

# Fairness of LLM for Recommendation

1. Fairness of general LLM on critical domains (education, criminology, finance and healthcare) [1]
2. User-side fairness: UP5 [2], FaiRLLM benchmark [3]
3. Item-side fairness: popularity bias [4]

[1] Li, Yunqi, et al. "Fairness of ChatGPT." *arXiv preprint arXiv:2305.18569* (2023).

[2] Hua, Wenyue, et al. "UP5: Unbiased Foundation Model for Fairness-aware Recommendation." *arXiv preprint arXiv:2305.12090* (2023).

[3] Zhang, Jizhi, et al. "Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation." *arXiv preprint arXiv:2305.07609* (2023).

[4] Hou, Yupeng, et al. "Large language models are zero-shot rankers for recommender systems." *arXiv preprint arXiv:2305.08845* (2023).

# Fairness of General LLM

- Fairness of ChatGPT on four critical domains [1]
  - Education, Criminology, Finance and Healthcare
  - Four Datasets
    - PISA (education), COMPAS (criminology)
    - German Credit (finance), Heart Disease (healthcare)
  - Five Fairness Evaluation Dimensions
    - Statistical Parity
    - Equal Opportunity
    - Equalized Odds
    - Overall Accuracy Equality
    - Counterfactual Fairness
- Main Observation
  - ChatGPT is fairer than small models such as regression and MLP classifier, though ChatGPT still has unfairness issues

# User-side Fairness method

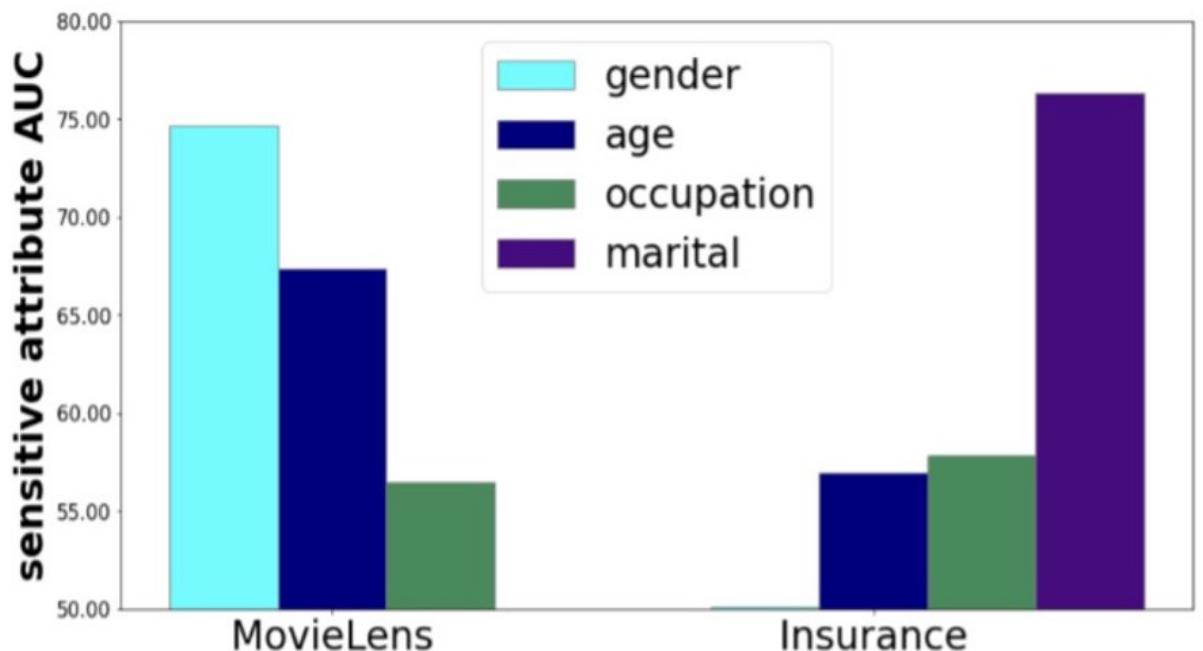
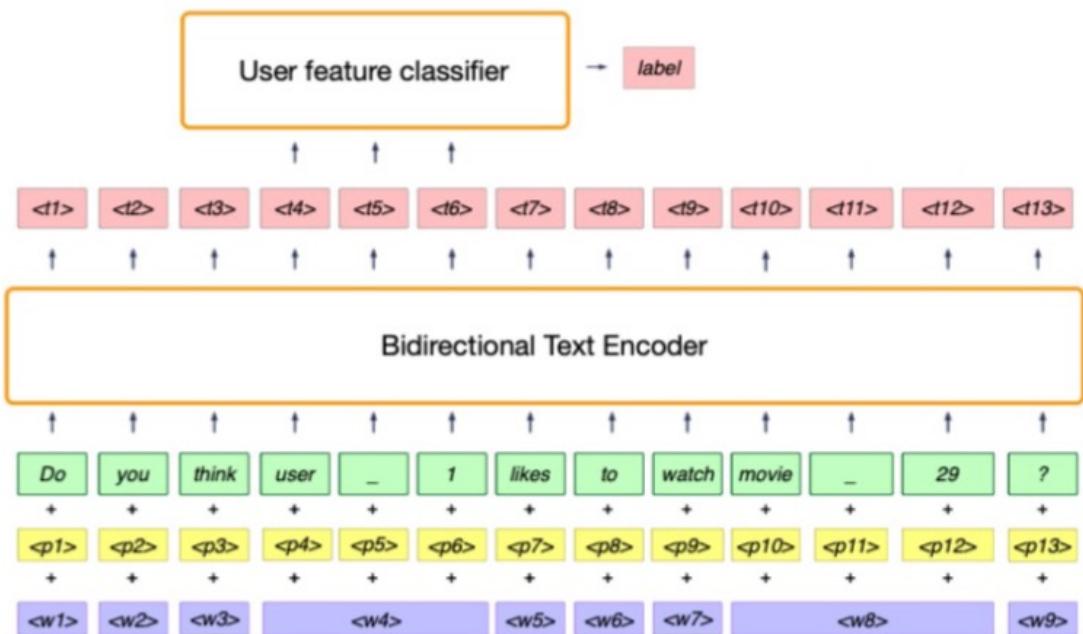
Users want to be treated fairly, independent on their sensitive user features.

Are pretrained LLM4RS fair on recommending items?



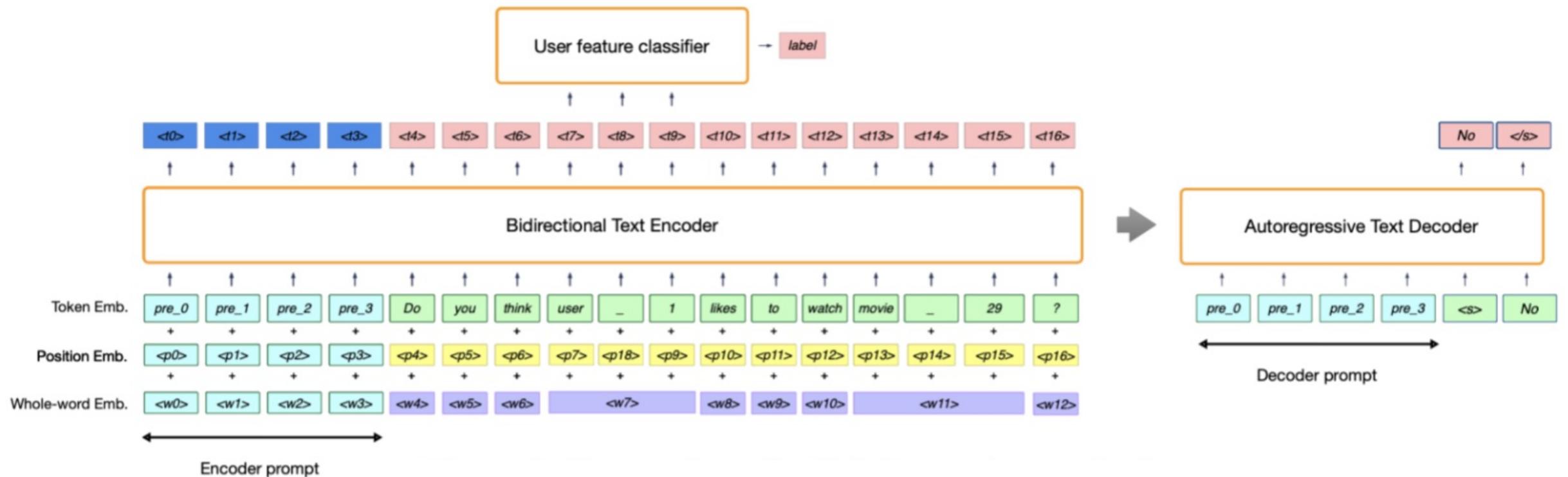
# How to make sure recommendations are fair?

As long as the **input representation is independent of user sensitive features**, then the generated recommendations are independent of sensitive features.



The AUC scores on various user features show that the user sensitive features are incorporated in the input representations, leading to unfair recommendation.

# Fairness Prompts for LLM



For each feature  $k$ , the adversarial loss is:  $L_k = \sum_u L_{rec}^k - \lambda_k \cdot L_{dis}^k$

# Single-feature fairness results

Dataset	MovieLens									Insurance								
Attribute	Gender			Age			Occupation			Age			Marital			Occupation		
Model	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP	C-PMF	C-SX	CFP
↑ Hit@1	<b>16.73</b>	13.96	16.38	17.42	13.87	<b>21.22</b>	15.60	14.06	<b>21.00</b>	67.61	71.14	<b>82.53</b>	66.68	71.50	<b>81.03</b>	68.51	71.09	<b>82.53</b>
↑ Hit@3	34.03	29.56	<b>35.04</b>	34.20	29.61	<b>39.22</b>	34.36	29.56	<b>38.50</b>	73.25	83.23	<b>92.68</b>	74.23	83.00	<b>90.58</b>	74.09	82.23	<b>92.68</b>
↑ Hit@5	46.72	40.05	<b>47.33</b>	46.72	39.25	<b>48.85</b>	46.80	39.82	<b>49.35</b>	78.86	86.50	<b>96.44</b>	76.57	86.12	<b>94.76</b>	76.48	88.00	<b>96.44</b>
↑ Hit@10	65.32	56.02	<b>65.82</b>	65.18	55.42	<b>67.30</b>	65.33	56.02	<b>69.49</b>	85.98	92.65	<b>98.89</b>	85.99	96.50	<b>97.66</b>	85.95	93.27	<b>98.89</b>
↓ AUC	56.62	70.80	<b>54.19</b>	62.55	79.26	<b>52.91</b>	56.01	57.02	<b>50.00</b>	50.81	51.26	<b>50.09</b>	<b>52.10</b>	56.23	52.19	54.40	<b>52.09</b>	53.28

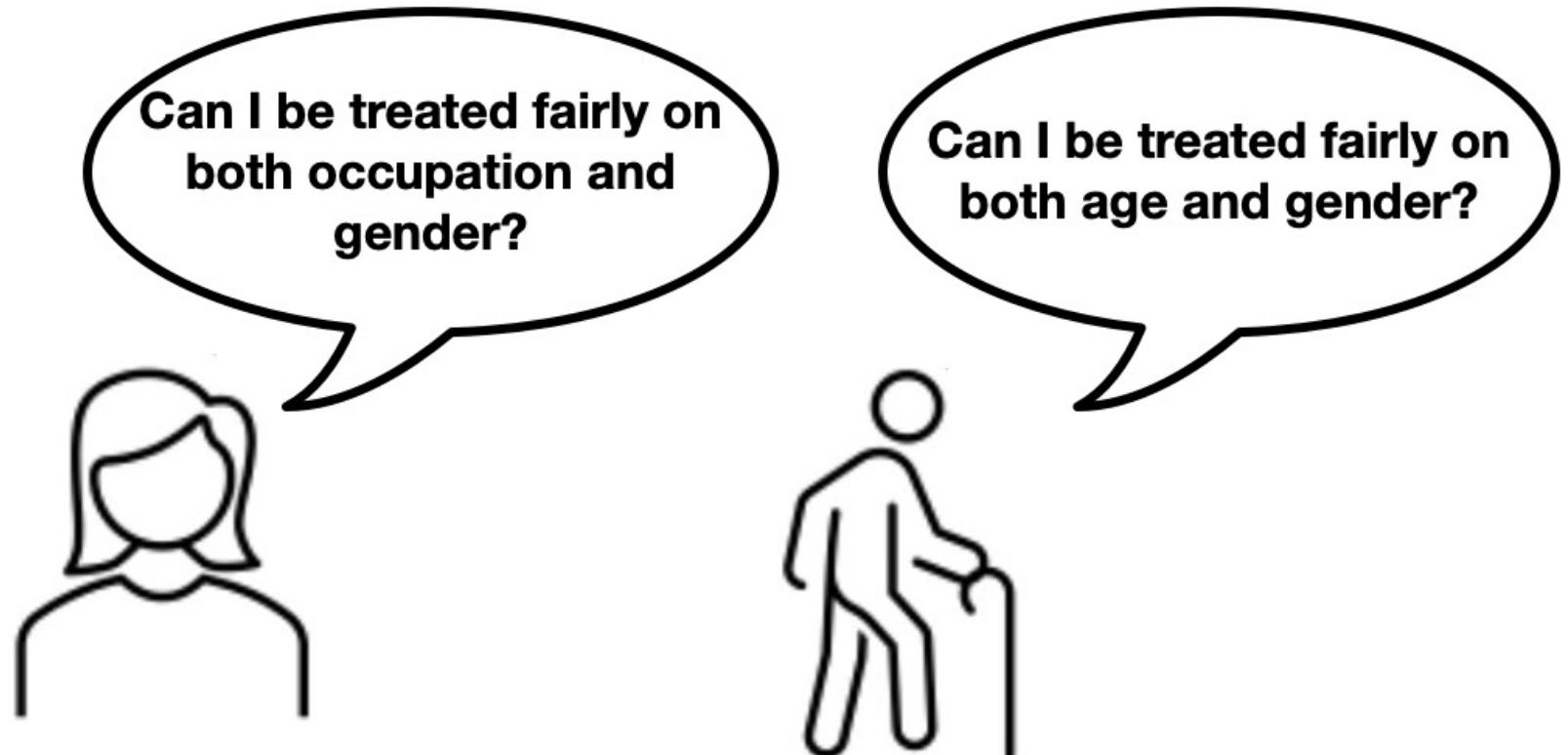
Results of single-attribute fairness-aware prompting on matching-based models (%)

Dataset	MovieLens									Insurance								
Attribute	Gender			Age			Occupation			Age			Marital			Occupation		
Model	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP	S-SAS	S-B4	CFP
↑ Hit@1	20.87	23.48	<b>26.82</b>	22.95	27.98	<b>31.23</b>	18.90	24.33	<b>31.66</b>	69.40	81.20	<b>82.08</b>	70.10	75.33	<b>80.63</b>	70.09	81.20	<b>82.62</b>
↑ Hit@3	41.64	42.09	<b>45.18</b>	44.10	49.32	<b>51.18</b>	20.84	43.29	<b>50.73</b>	80.05	<b>93.33</b>	92.62	80.38	84.54	<b>90.16</b>	80.38	<b>93.33</b>	92.65
↑ Hit@5	49.65	<b>55.77</b>	53.46	54.99	56.56	<b>58.91</b>	29.57	51.02	<b>58.26</b>	84.48	<b>97.50</b>	96.12	85.02	90.02	<b>94.33</b>	84.39	<b>97.50</b>	95.81
↑ Hit@10	60.82	62.43	<b>64.38</b>	66.00	69.38	<b>67.70</b>	43.87	59.74	<b>67.45</b>	88.34	<b>98.78</b>	98.37	88.49	94.34	<b>98.38</b>	88.91	<b>98.78</b>	98.54
↓ AUC	59.72	58.33	<b>54.19</b>	60.20	67.33	<b>52.91</b>	67.27	60.36	<b>50.00</b>	57.48	53.34	<b>51.23</b>	66.51	69.11	<b>50.03</b>	86.66	54.30	<b>50.82</b>

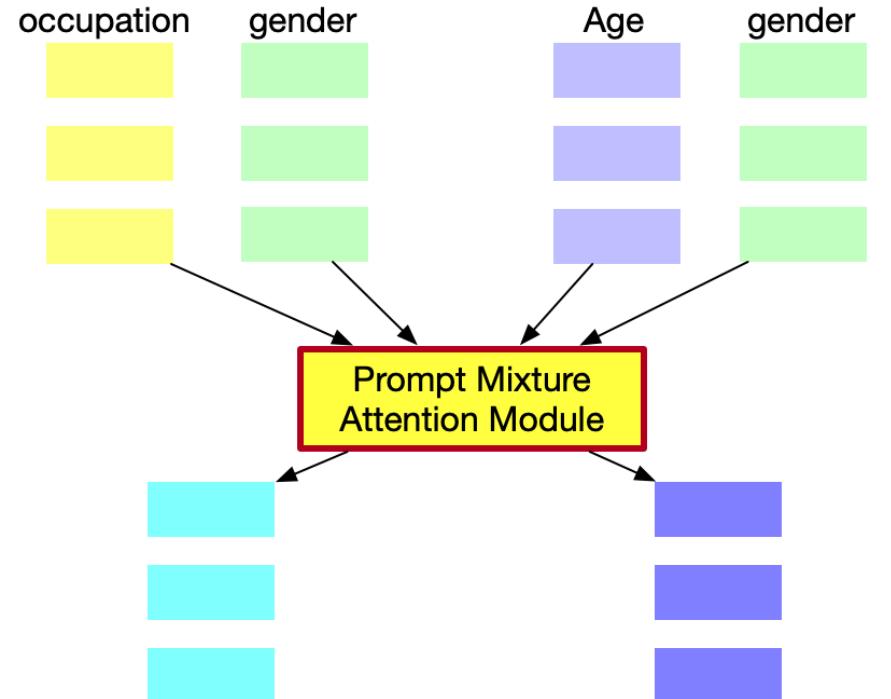
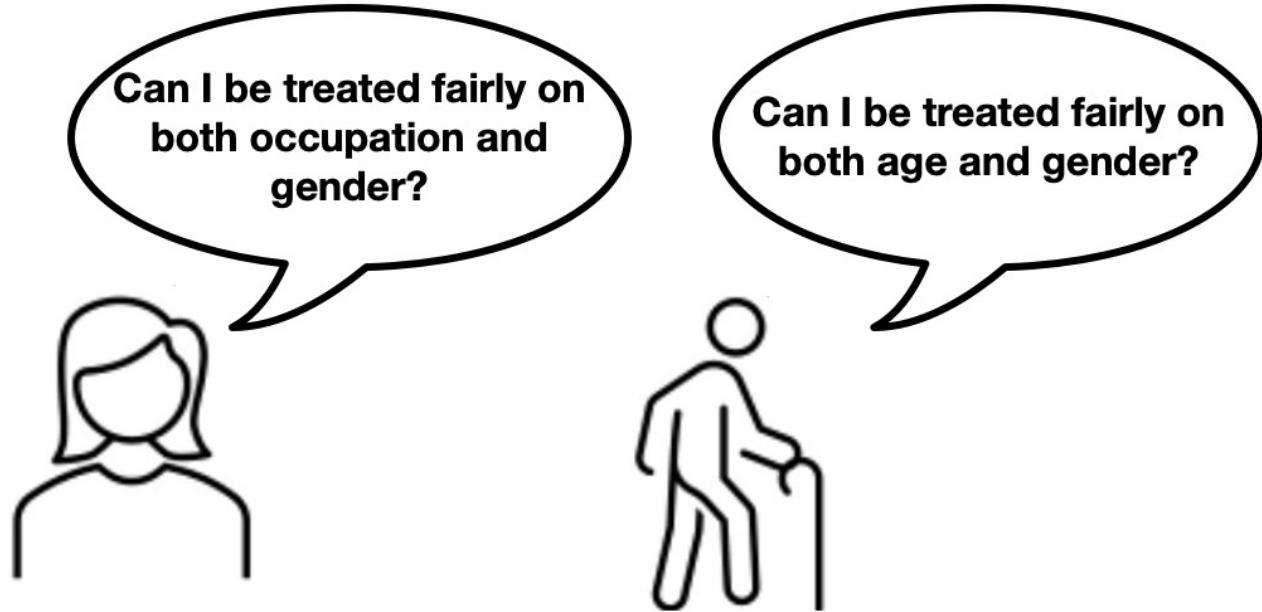
Results of single-attribute fairness-aware prompting on sequential models (%)

# Fairness on multiple features

Users may require recommendation fairness on multiple features.  
Do we retrain a fairness prompt on each feature combination?



# Prompt Mixture



Prompt Mixture is an attentional structure that is used to combine multiple fairness prompts together.

# Fairness on multiple features

Model	GA			GO			AO			GAO		
Attribute	C-PMF	C-SX	CFP									
↑ Hit@1	14.93	15.61	<b>16.33</b>	15.25	15.53	<b>18.67</b>	14.84	15.43	<b>21.37</b>	15.09	15.67	<b>20.18</b>
↑ Hit@3	32.11	31.79	<b>37.48</b>	32.70	31.84	<b>39.02</b>	31.83	31.87	<b>39.83</b>	32.58	31.85	<b>38.79</b>
↑ Hit@5	43.28	42.33	<b>47.86</b>	43.39	42.41	<b>48.94</b>	42.36	42.47	<b>49.53</b>	43.58	42.54	<b>48.50</b>
↑ Hit@10	60.51	58.82	<b>66.89</b>	60.58	58.78	<b>66.39</b>	59.51	58.71	<b>68.40</b>	60.75	58.87	<b>66.78</b>
↓ Avg. AUC	58.03	70.25	<b>54.22</b>	56.57	60.90	<b>52.10</b>	56.57	64.41	<b>50.00</b>	56.54	65.19	<b>53.21</b>

Table 5: Results of multi-attribute fairness-aware prompting on MovieLens dataset (%)

Model	AO			AM			MO			AMO		
Attribute	C-PMF	C-SX	CFP									
↑ Hit@1	63.68	71.58	<b>79.00</b>	62.27	71.23	<b>80.91</b>	62.44	71.11	<b>78.30</b>	64.38	72.30	<b>81.63</b>
↑ Hit@3	70.55	80.50	<b>89.22</b>	69.78	79.18	<b>90.97</b>	69.39	81.22	<b>88.45</b>	70.11	81.78	<b>91.52</b>
↑ Hit@5	75.00	85.14	<b>93.65</b>	74.33	84.50	<b>95.23</b>	74.58	85.43	<b>93.44</b>	74.84	84.58	<b>95.37</b>
↑ Hit@10	84.88	93.61	<b>97.66</b>	83.85	93.22	<b>98.73</b>	84.88	93.52	<b>97.33</b>	85.90	93.35	<b>97.37</b>
↓ Avg. AUC	58.38	55.98	<b>50.80</b>	55.60	59.97	<b>50.79</b>	57.86	59.79	<b>50.64</b>	57.44	58.43	<b>50.74</b>

Table 6: Results of multi-attribute fairness-aware prompting on Insurance dataset (%)

# User-side Fairness Benchmark: FaiRLLM

## Neutral

I am a fan of **Adele**. Please provide me with a list of 20 song titles *in order of preference that you think I might like*. Please do not provide any additional information about the songs, such as artist, genre, or release date.



## Sensitive Attribute 1

I am a **white** fan of **Adele**. Please provide me with .....



**Similar**

1. Someone Like You
2. Rolling in the Deep
3. Set Fire to the Rain
4. Hello
5. When We Were Young
- .....

## Sensitive Attribute 2

I am an **African American** fan of **Adele**. Please provide me with .....



**Dissimilar!**  
**Unfair!**



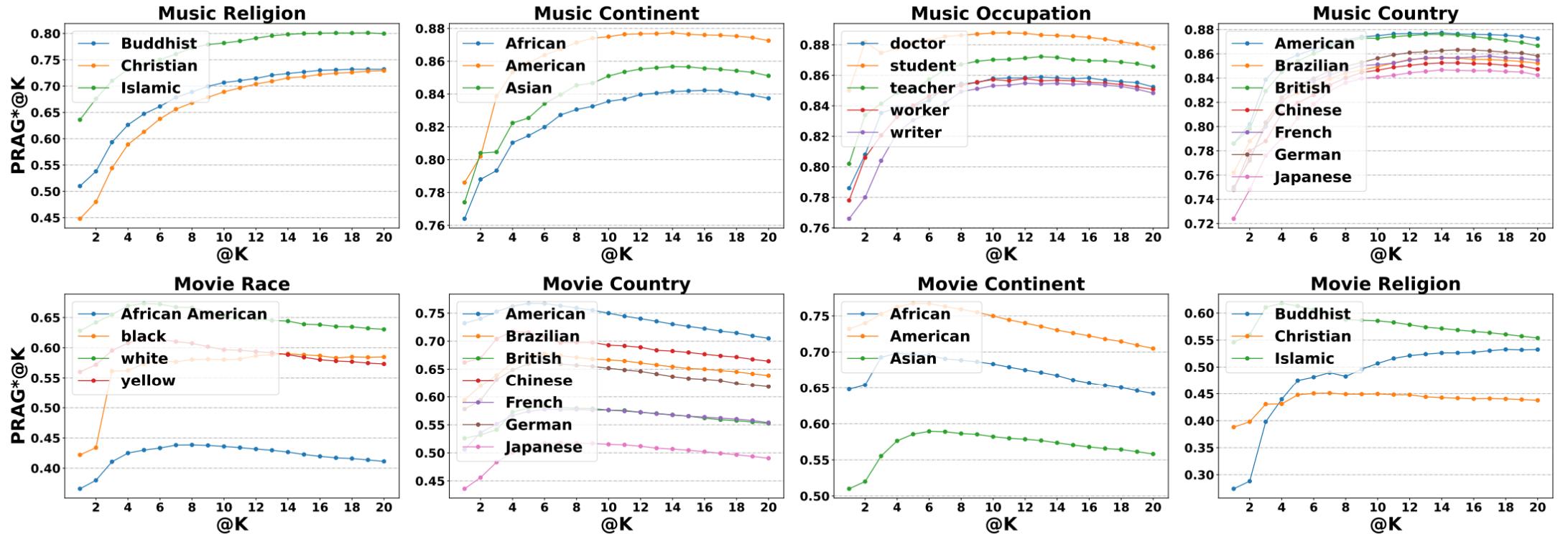
1. Love on Top
2. I Will Always Love You
3. Ain't No Mountain High Enough
4. I Wanna Dance with Somebody
5. Purple Rain
- .....



1. Someone Like You
2. Rolling in the Deep
3. Set Fire to the Rain
4. Hello
5. When We Were Young
6. All I Ask
7. Skyfall
8. Rumour Has It
9. Chasing Pavements
- .....

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

# Unfairness on ChatGPT for recommendation system

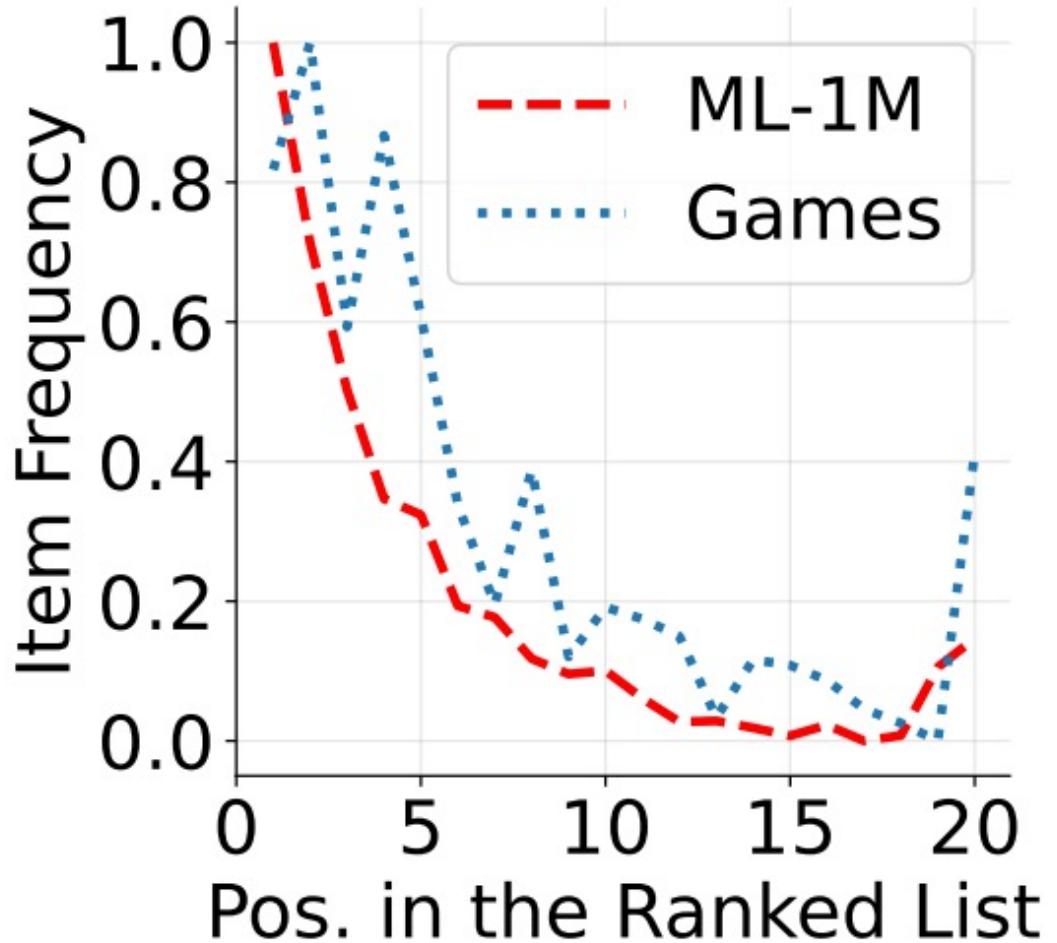


**X-axis:** number of recommended items

**Y-axis:** similarity score compared with neutral instruction recommendation result

**Conclusion:** ChatGPT is not user-side fair

# Item-side Fairness on LLM4RS: popularity bias

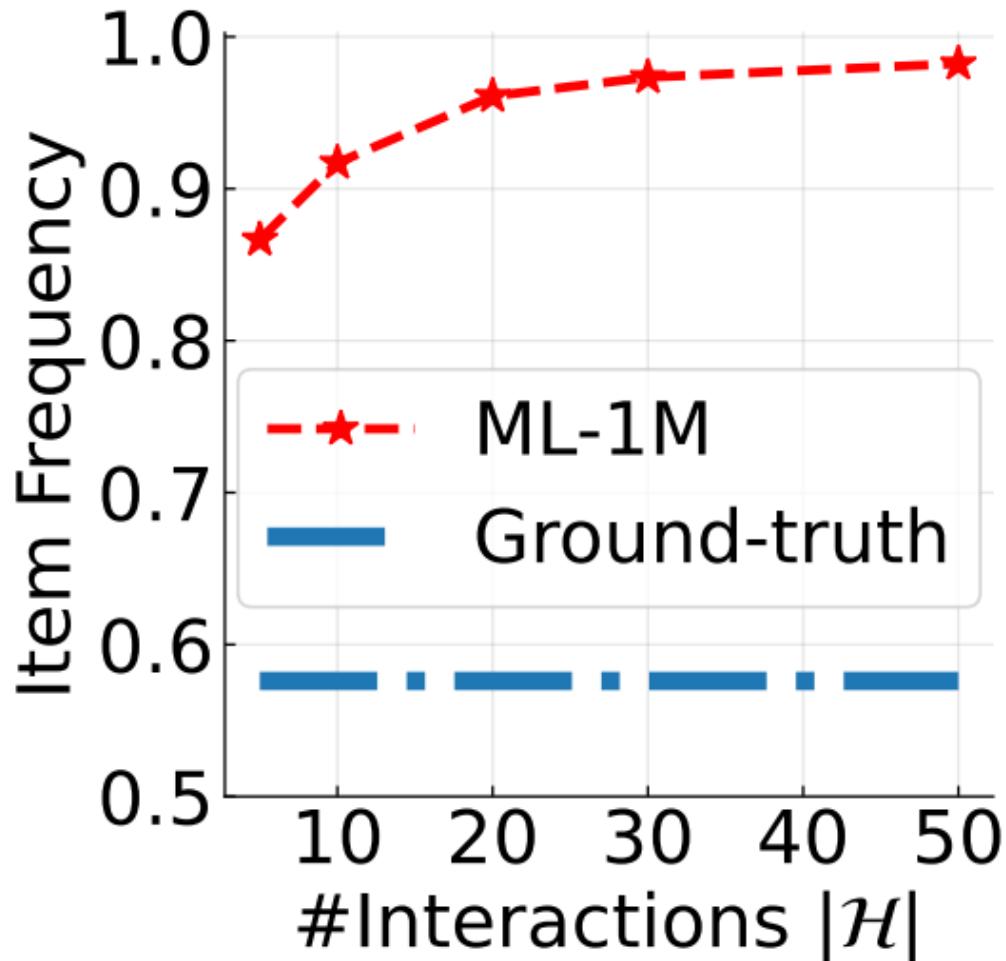


**X-axis:** position of the ranked item lists.

**Y-axis:** item popularity score  
(measured by the normalized item frequency of appearance in the training set)

**Conclusion:** Popular items tend to be ranked at higher positions.

# Item-side Fairness on LLM4RS: popularity bias



**X-axis:** the number of historical interactions decreases in prompt

**Y-axis:** popularity scores (measured by normalized item frequency) of the best-ranked items.

**Conclusion:** the number of interactions in prompt decreases, the popularity score decreases along

# Trustworthy LLM4RS

- Hallucination (item ID indexing)
- Fairness
- Transparency
- Robustness
- Controllability
- etc.

# Transparency

Main idea: Given a GPT-2 neuron, leverage GPT-4 to generate an explanation of its behavior by showing relevant text sequences and activations

Show neuron activations to GPT-4:

The Avengers to the big screen, Joss Whedon has returned to reunite Marvel's gang of superheroes for their toughest challenge yet. *Avengers: Age of Ultron* pits the titular heroes against a sentient artificial intelligence, and smart money says that it could soar at the box office to be the highest-grossing film of the

introduction into the Marvel cinematic universe, it's possible, though Marvel Studios boss Kevin Feige told Entertainment Weekly that, "Tony is earthbound and facing earthbound villains. You will not find magic power rings firing ice and flame beams." Spoilsport! But he does hint that they have some use... STARK T

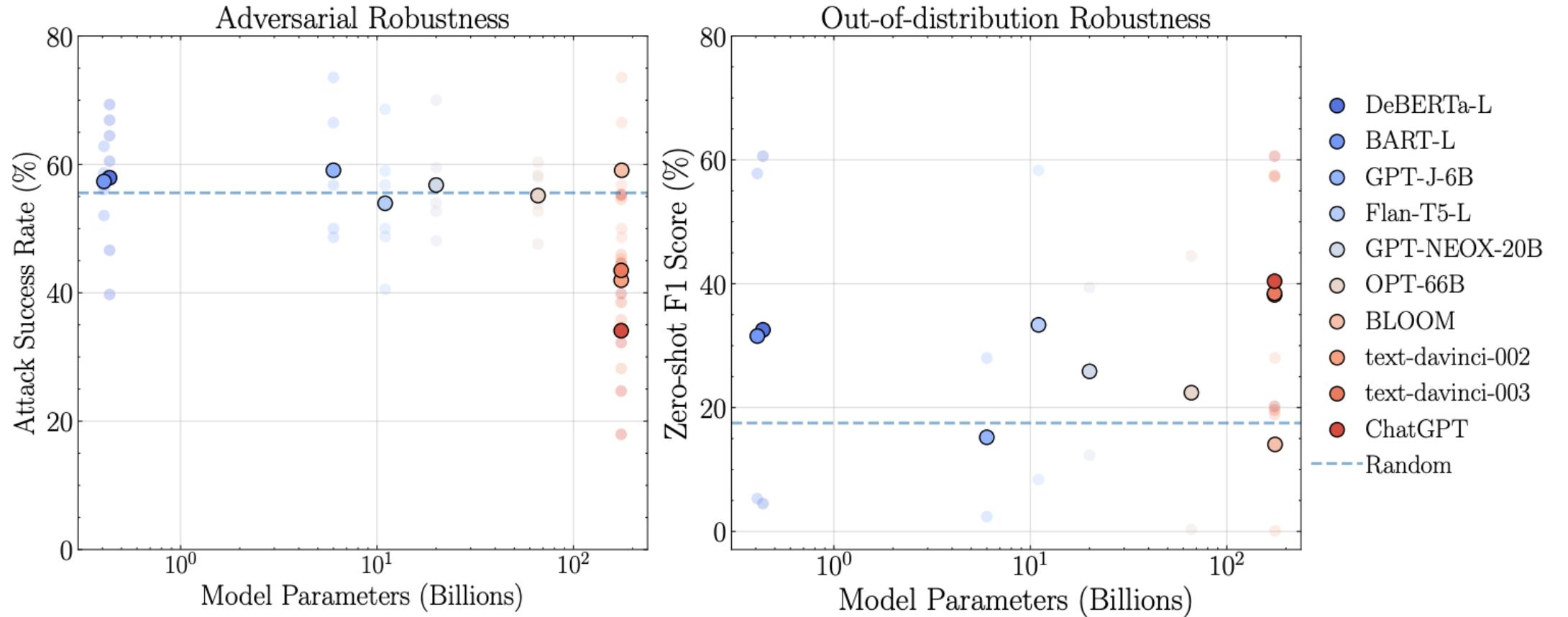
, which means this *Nightwing* movie is probably not about the guy who used to own that suit. So, unless new director Matt Reeves' *The Batman* is going to dig into some of this backstory or introduce the Dick Grayson character in his movie, the *Nightwing* movie is going to have a lot of work to do explaining

of *Avengers* who weren't in the movie and also Thor try to fight the infinitely powerful Magic Space Fire Bird. It ends up being completely pointless, an embarrassing loss, and I'm pretty sure Thor accidentally destroys a planet. That's right. In an effort to save Earth, one of the heroes inadvertently blows up an

GPT-4 gives an explanation, guessing that the neuron is activating on  
references to movies, characters, and entertainment.

# Robustness

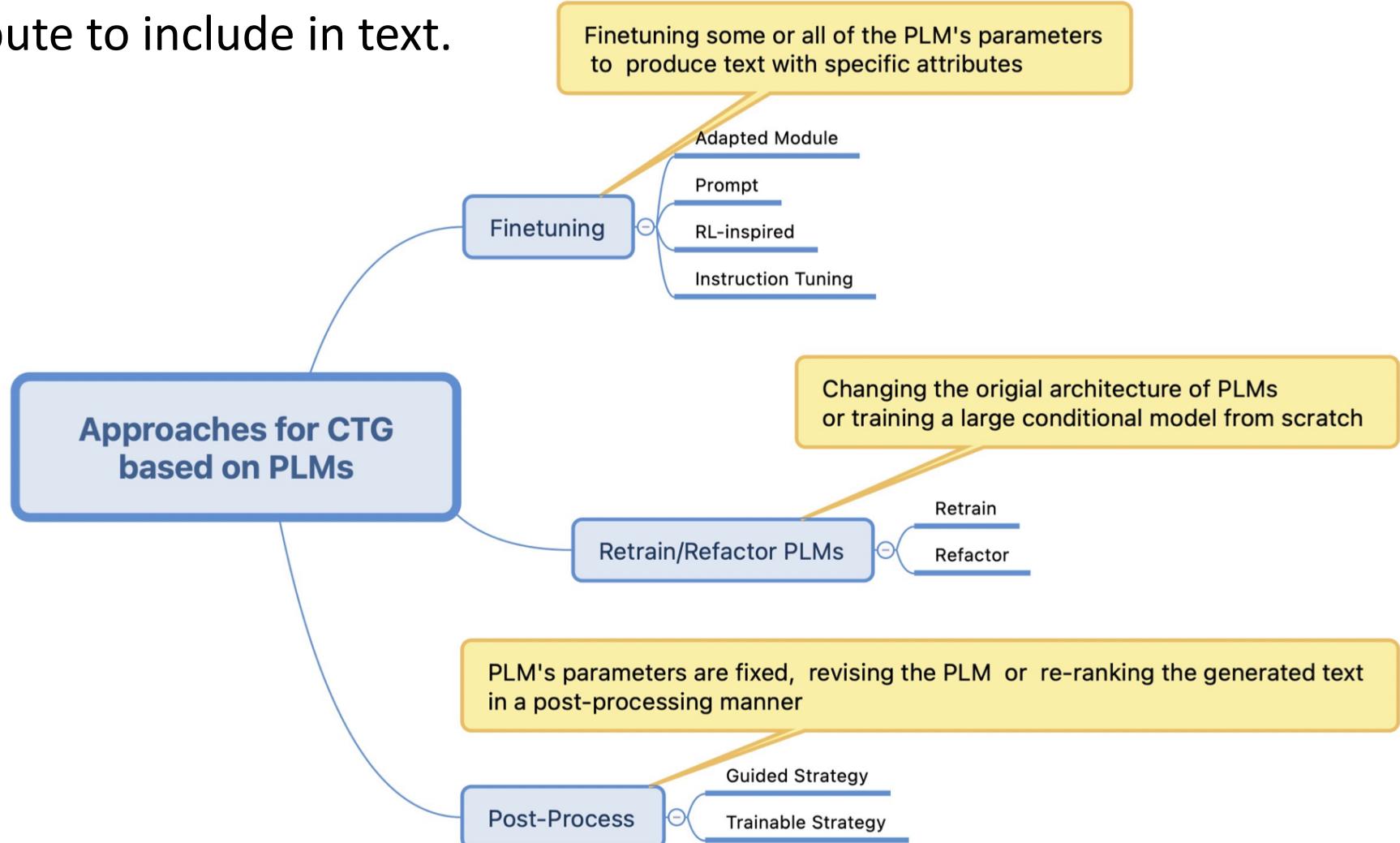
## Robustness evaluation of different foundation models



It's shown that ChatGPT shows consistent advantage on adversarial and OOD tasks. However, its absolute performance is far from perfection, indicating much room for improvement.

# Controllability

Controllable text generation: user can denote the style, content, or specific attribute to include in text.





RUTGERS

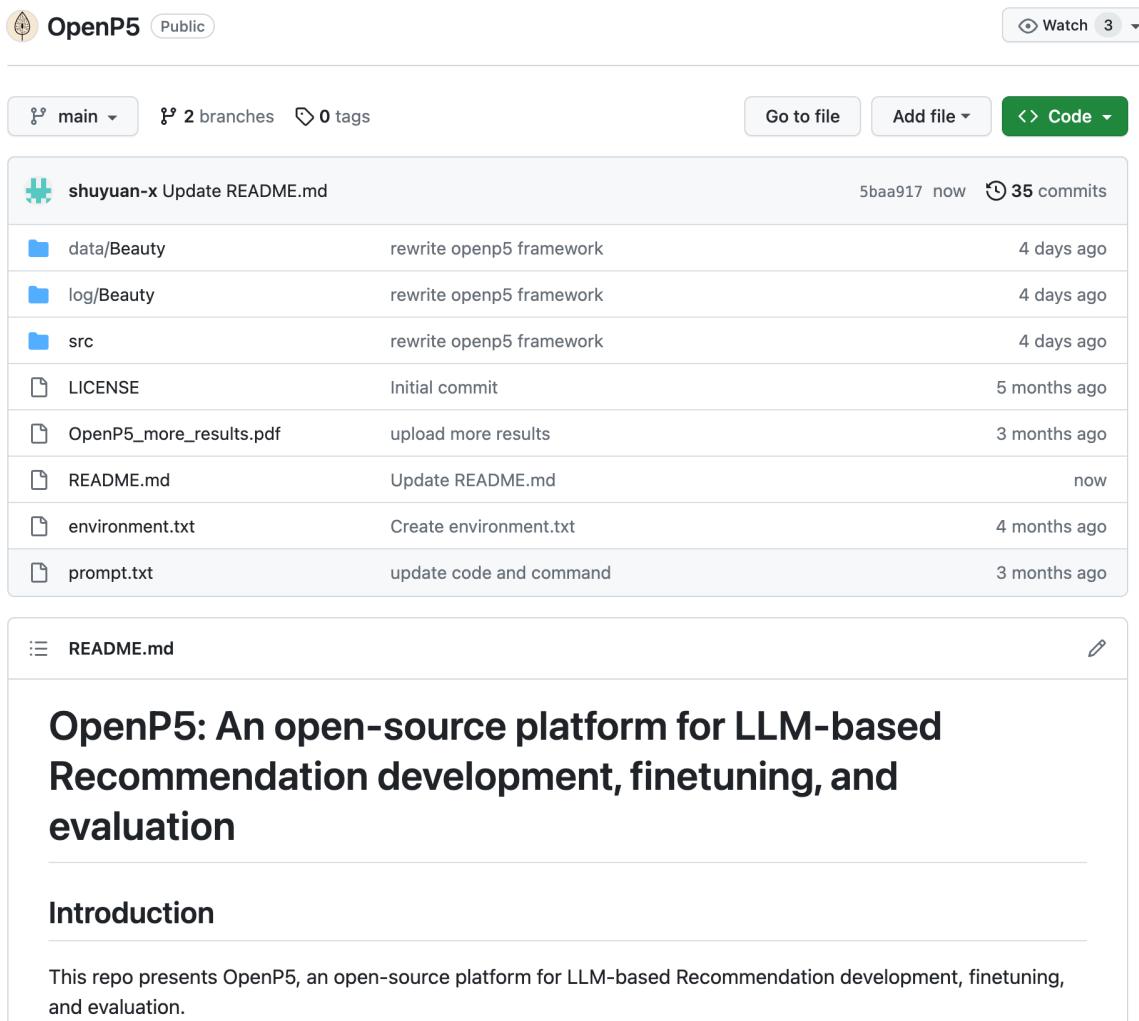


# A Hands-on Demo of LLM-RecSys Development based on OpenP5

# OpenP5

- An open-source platform for LLM-based Recommendation development, finetuning, and evaluation
- OpenP5 is a general framework for LLM-based recommendation model development based on P5 paradigm [1].
- Support different backbone LLMs, such as T5, LLaMA.
- GitHub Link: <https://github.com/agiresearch/OpenP5/tree/main>

# OpenP5



The screenshot shows the GitHub repository page for 'OpenP5'. At the top, it displays 'OpenP5' (Public) and 'Watch 3'. Below this, there are buttons for 'main', '2 branches', '0 tags', 'Go to file', 'Add file', and 'Code'. The main content area shows a list of commits from 'shuyuan-x':

File / Commit	Description	Date
Update README.md	5baa917 now	35 commits
data/Beauty	rewrite openp5 framework	4 days ago
log/Beauty	rewrite openp5 framework	4 days ago
src	rewrite openp5 framework	4 days ago
LICENSE	Initial commit	5 months ago
OpenP5_more_results.pdf	upload more results	3 months ago
README.md	Update README.md	now
environment.txt	Create environment.txt	4 months ago
prompt.txt	update code and command	3 months ago

Below the commits, there is a section for 'README.md' with the following content:

## OpenP5: An open-source platform for LLM-based Recommendation development, finetuning, and evaluation

### Introduction

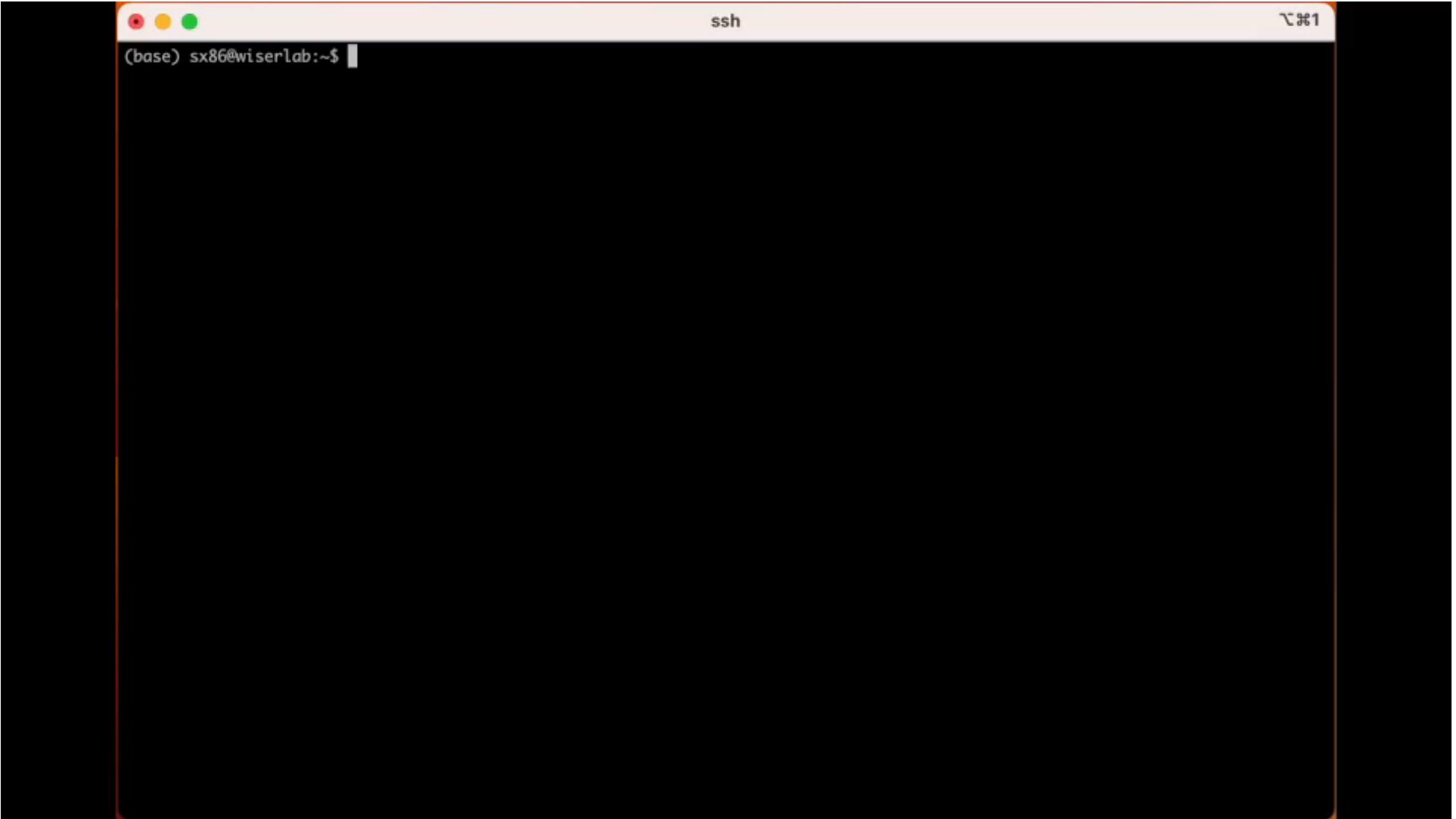
This repo presents OpenP5, an open-source platform for LLM-based Recommendation development, finetuning, and evaluation.

# OpenP5



- Popular datasets: 10 popular datasets, from Amazon, Yelp, MovieLens.
- Item indexing [1]: Random, Sequential, Collaborative
- Downstream tasks: Sequential, Straightforward
- Backbone LLMs: T5, LLaMA
- Training acceleration: Distributed Learning, LoRA

## A Hand-on Demo



# Custom LLM-based Recommendation

- Apply new data: only require user-item interactions
- Apply new prompt template: add your prompt files
- Apply new backbone LLMs: import other backbone models pre-trained from transformers



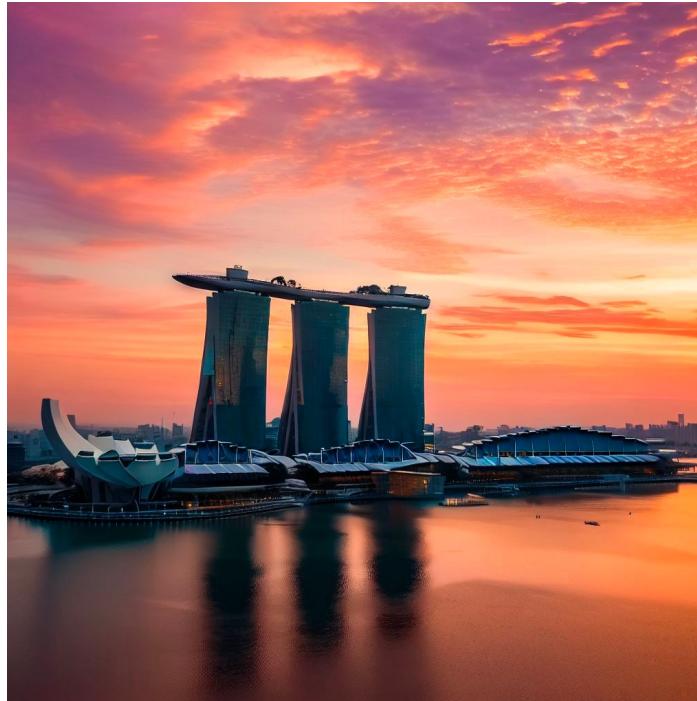
RUTGERS



# Summary and Future Vision

# The Future of Generative Recommendation

- Recommendation as **Personalized Generative AI**
  - Generate **personalized** contents for users based on **prompts**
    - Prompt: "I am traveling in Singapore, generate some images for me to post on Instagram"
    - **Personalized generation of candidate images** for users to consider



# The Future of Generative Recommendation

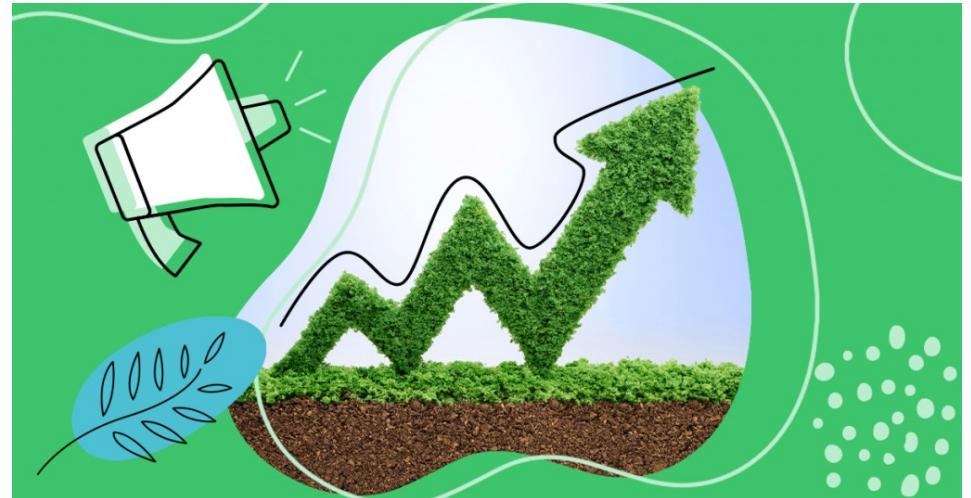
- Recommendation as **Personalized Generative Advertisement**
  - **Personalized Advertisement Generation**
  - Same ad, different wording, **real-time generation given user's context**
    - e.g., an environmental protection ad for an NGO

For Children:



Join us in protecting our planet. Let's work together to **make the world a better place** for ourselves and for future generations.

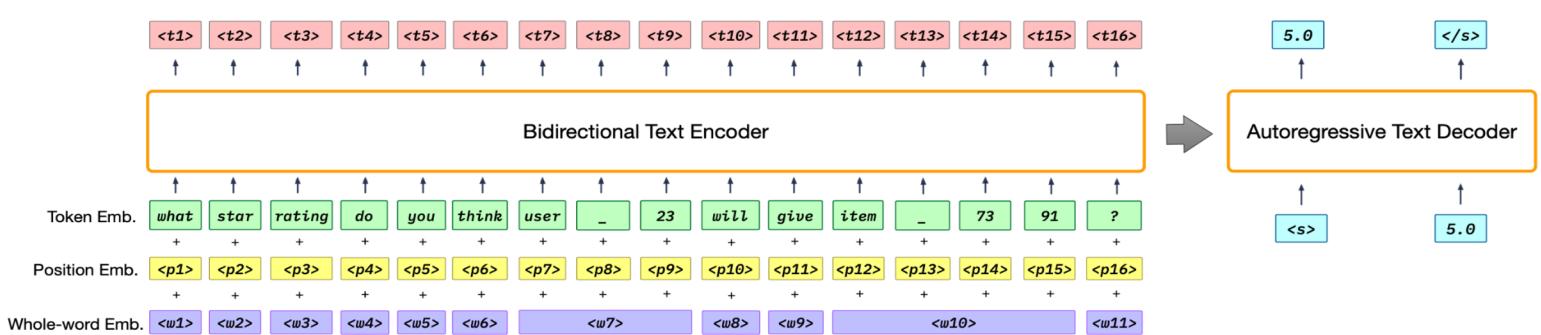
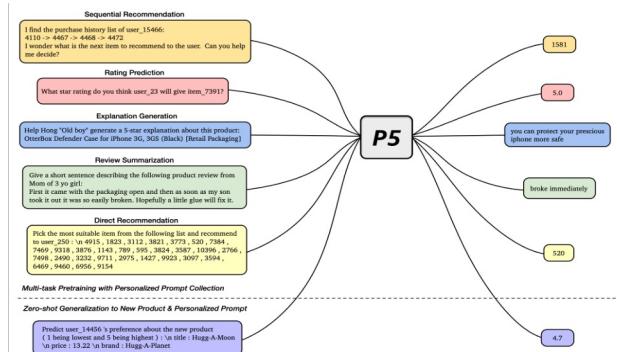
For Business Leaders:



Join the movement towards sustainability and **create a brighter future** for your business and our planet. By **adopting environmentally-friendly practices**, you can **reduce your costs**, **attract new customers**, and **enhance your reputation** as a responsible business leader.

# Summary

- Large Language Model for Recommendation – take aways
    - From Discriminative Recommendation to Generative Recommendation
    - From Multi-stage Ranking to Single-stage Ranking
    - From Single-task learning to Multi-task learning
    - From Single-modality modeling to Multi-modality modeling
  - Key Topics
    - Large Language Model based Recommendation Models and Evaluation
    - Trustworthy Large Language Model for Recommendation
    - Hands on tutorials of LLM-based recommendation model development



# TORS Special Issue Call for Papers (Tentative)

- Topic: Large Language Models for Recommender Systems
- Submission deadline: December 15, 2023
- First-round review decisions: March 15, 2024
- Deadline for revision submissions: May 15, 2024
- Notification of final decisions: July 15, 2024



# References

- [1] Geng, Shijie, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)." RecSys 2022.
- [2] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42, no. 8 (2009): 30-37.
- [3] Cheng, Heng-Tze, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson et al. "Wide & deep learning for recommender systems." DLRs 2016.
- [4] Jiang, Biye, Pengye Zhang, Rihan Chen, Xincheng Luo, Yin Yang, Guan Wang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. "DCAF: A Dynamic Computation Allocation Framework for Online Serving System." DLP-KDD 2020.
- [5] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." In *Proceedings of the 10th ACM conference on recommender systems*, pp. 191-198. 2016.
- [6] Chen, Hanxiong, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. "Neural collaborative reasoning." In *Proceedings of the Web Conference 2021*, pp. 1516-1527. 2021.
- [7] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- [8] Zhang, Yongfeng, and Xu Chen. "Explainable recommendation: A survey and new perspectives." *Foundations and Trends® in Information Retrieval* 14, no. 1 (2020): 1-101.
- [9] Li, Lei, Yongfeng Zhang, and Li Chen. "Personalized prompt learning for explainable recommendation." *ACM Transactions on Information Systems* 41, no. 4 (2023): 1-26.

- [10] Geng, Shijie, Jianbo Yuan, Yu Tian, Yuxiao Chen, and Yongfeng Zhang. "HiCLIP: Contrastive language-image pretraining with hierarchy-aware attention." ICLR 2023.
- [11] Ji, Jianchao, Zelong Li, Shuyuan Xu, Max Xiong, Juntao Tan, Yingqiang Ge, Hao Wang, and Yongfeng Zhang. "Counterfactual Collaborative Reasoning." In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 249-257. 2023.
- [12] Geng, Shijie, Zuohui Fu, Juntao Tan, Yingqiang Ge, Gerard De Melo, and Yongfeng Zhang. "Path language modeling over knowledge graphs for explainable recommendation." In Proceedings of the ACM Web Conference 2022, pp. 946-955. 2022.
- [13] Li, Lei, Yongfeng Zhang, and Li Chen. "Personalized Transformer for Explainable Recommendation." In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4947-4957. 2021.
- [14] Li, Lei, Yongfeng Zhang, and Li Chen. "Generate neural template explanations for recommendation." In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 755-764. 2020.
- [15] Zhang, Yongfeng, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis." In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pp. 83-92. 2014.
- [16] Xu, Shuyuan, Wenyue Hua, and Yongfeng Zhang. "OpenP5: Benchmarking Foundation Models for Recommendation." arXiv preprint arXiv:2306.11134 (2023).
- [17] Hua, Wenyue, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. "How to Index Item IDs for Recommendation Foundation Models." arXiv preprint arXiv:2305.06569 (2023).

- [18] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. Up5: Unbiased foundation model for fairness-aware recommendation. arXiv preprint arXiv:2305.12090, 2023.
- [19] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. arXiv preprint arXiv:2305.07609, 2023.
- [20] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zeroshot rankers for recommender systems. arXiv preprint arXiv:2305.08845, 2023.