THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

NYU SHANGHAI 上海纽约大学

IJCAI JEJU 2024

Recommender Systems in the Era of Large Language Models (LLMs)

Yujuan Ding[1], Shijie Wang[1] , Liangbo Ning[1] , Qiaoyu Tan[2] , Wenqi Fan[1], Qing Li[1]

[1]The Hong Kong Polytechnic University
[2]New York University (Shanghai)

Survey

August 4th (Day 2), Afternoon 1 & 2
IJCAI 2024, Jeju, Korea
Zoom ID: *864 7573 0054*, Password: *732469*

Website

Website (Slides): https://advanced-recommender-systems.github.io/LLMs4Rec-IJCAI/
Survey Paper: "Recommender Systems in the Era of Large Language Models (LLMs)." TKDE 2024. https://arxiv.org/abs/2307.02046

Tutorial Outline

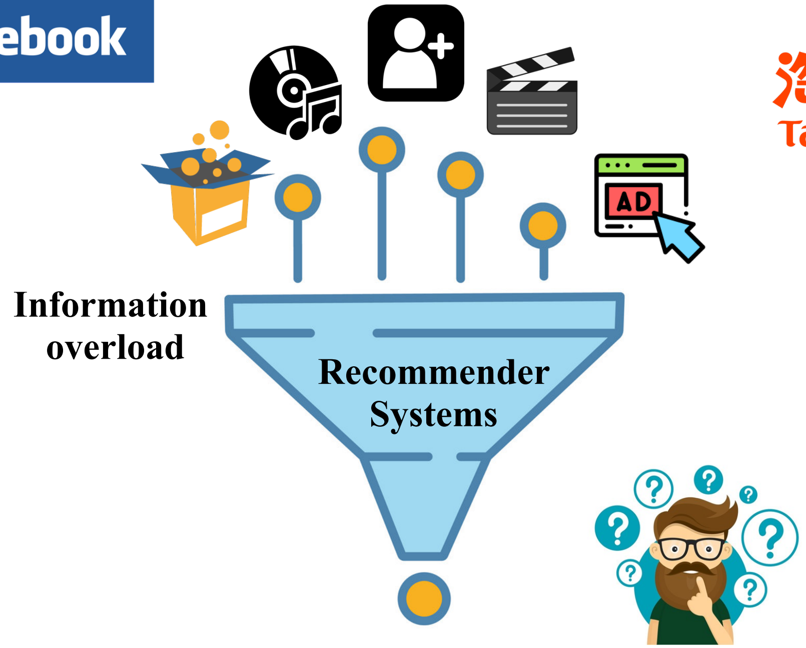
- **Part 1: Introduction of RecSys in the era of LLMs (Dr. Wenqi Fan)**
- **Part 2: Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3: Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4: Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5: Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 5: Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website

Recommender Systems (RecSys)

Age of Information Explosion



Recommend item X to user

Items can be: Products, Friends, News, Movies, Videos, etc.

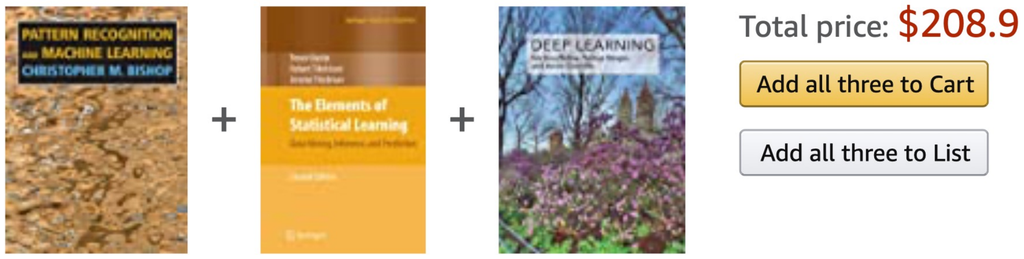
Recommender Systems (RecSys)

- Recommendation has been widely applied in online services:
  - E-commerce, Content Sharing, Social Networking ...

amazon ebay 淘宝网 Taobao.com JD.COM

Product Recommendation

Frequently bought together



A B C

Amazon's recommendation algorithm drives **35%** of its sales [from McKinsey, 2012]

Recommender Systems (RecSys)

- Recommendation has been widely applied in online services:
  - E-commerce, Content Sharing, Social Networking ...

YouTube

TikTok

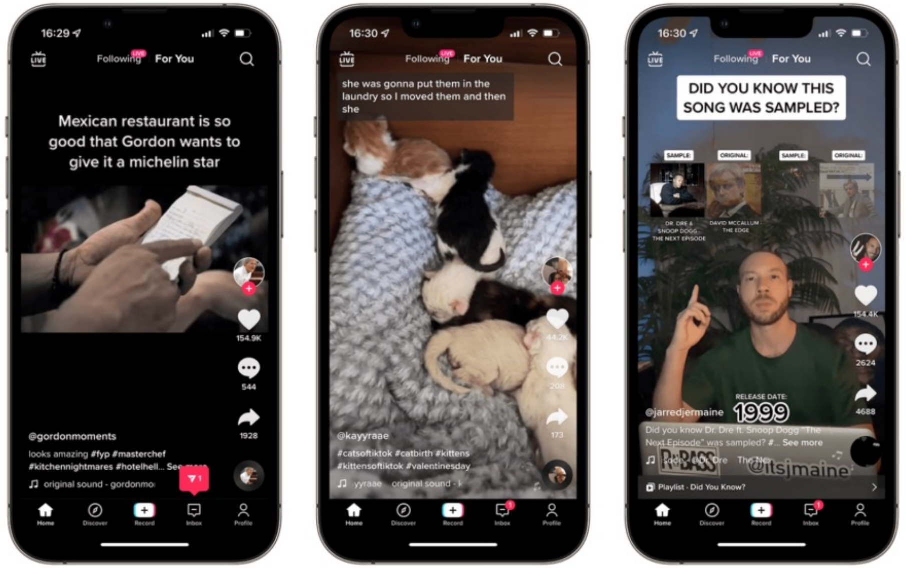
Google News

News/Video/Image Recommendation

<u>TikTok's recommendation algorithm</u>

Top 10 Global Breakthrough Technologies in 2021

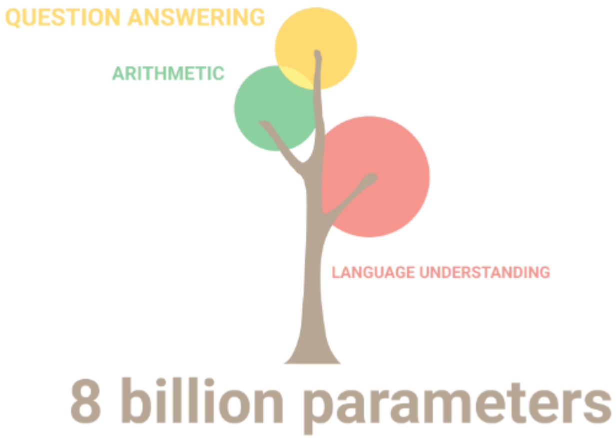
MIT Technology Review



Large Language Models (LLMs)

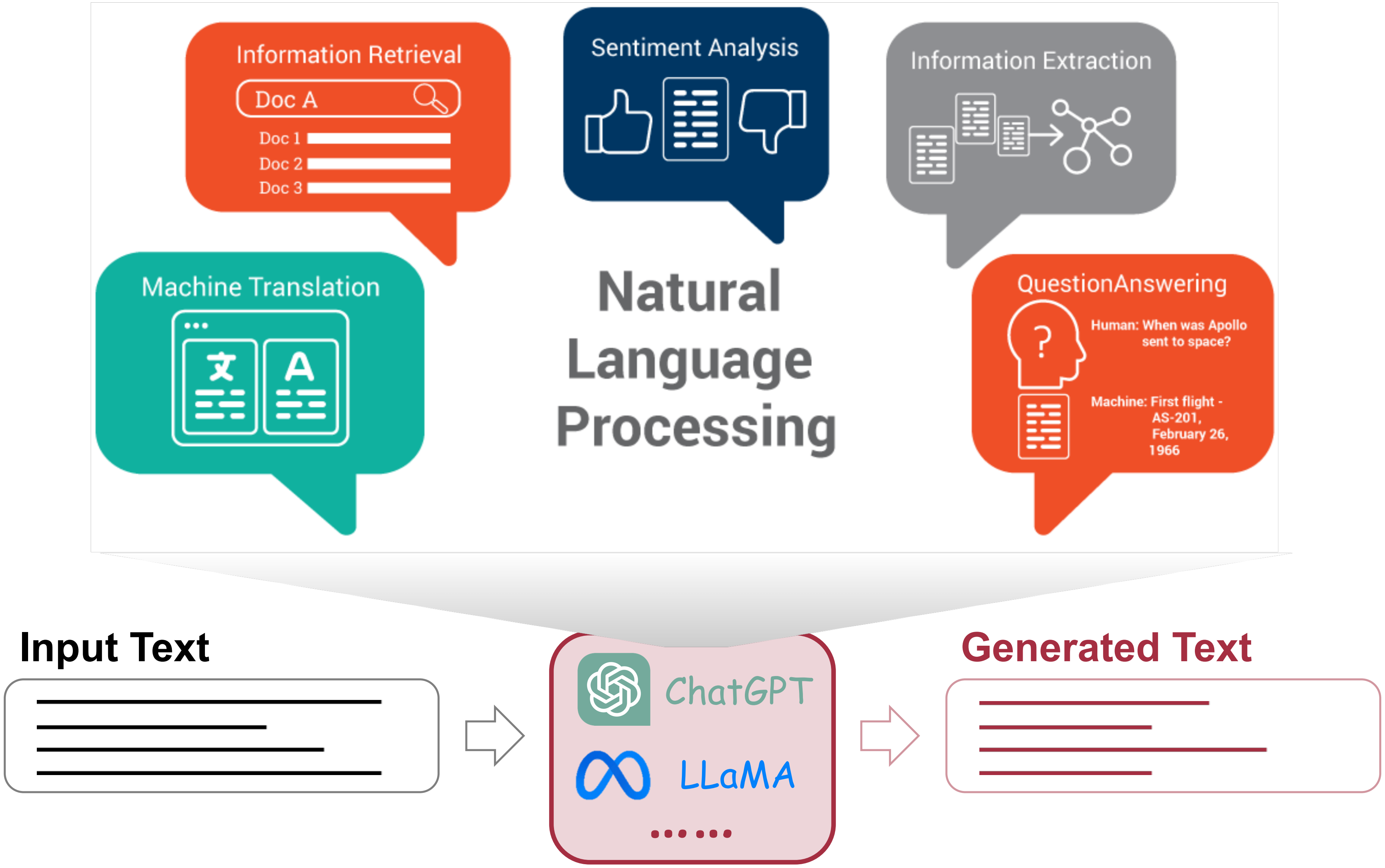
They Are Changing Our Lives !

OpenAI | Meta | Google AI | T5 | ......



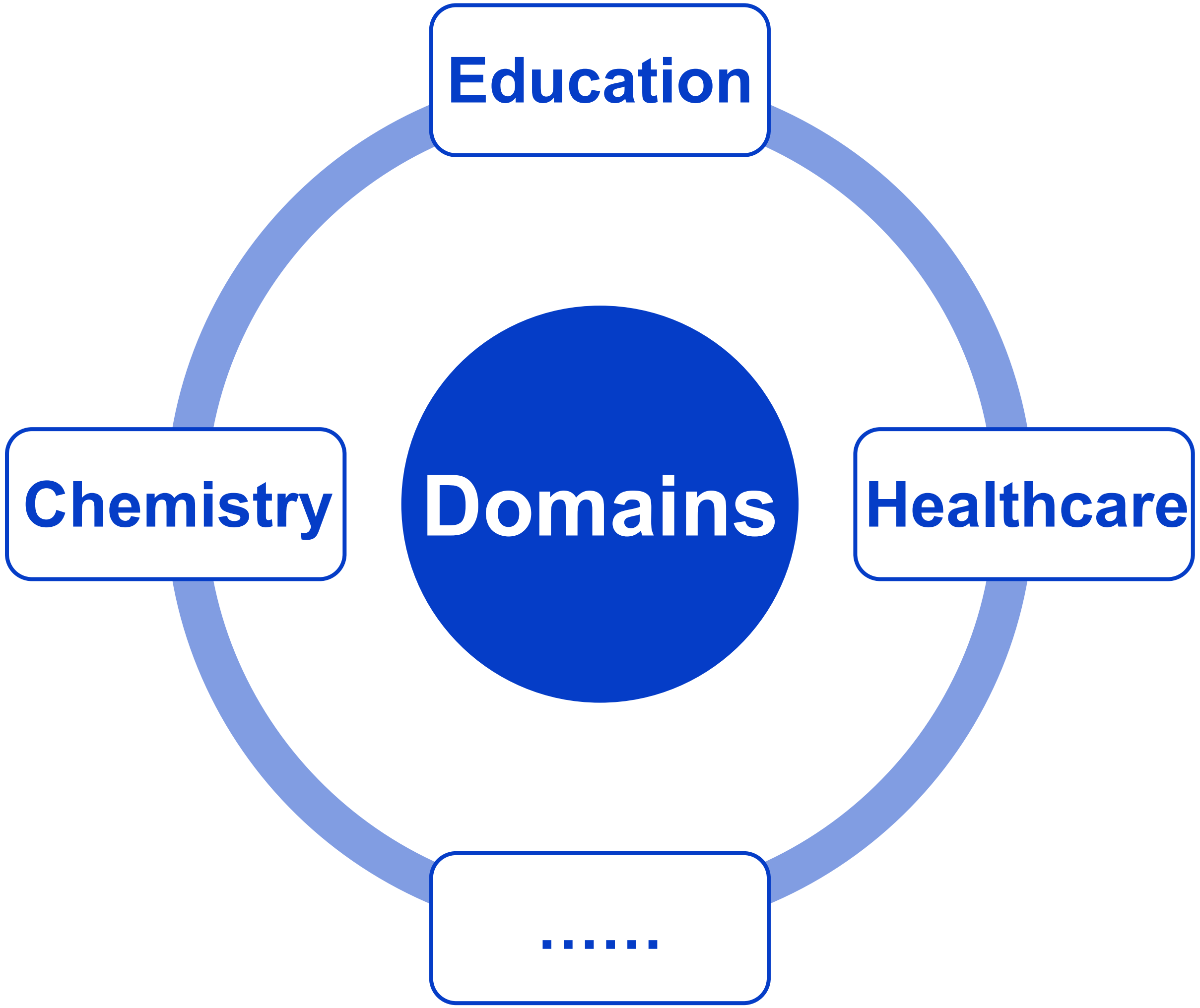
https://github.com/Hannibal046/Awesome-LLM/tree/main

LLMs in Natural Language Processing

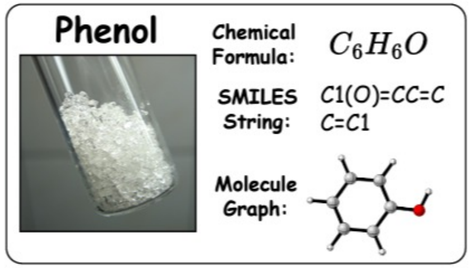


Large Language Models (LLMs)

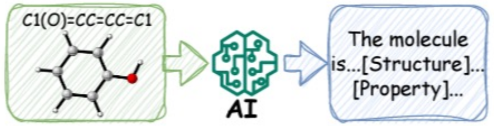
LLMs in Downstream Domains



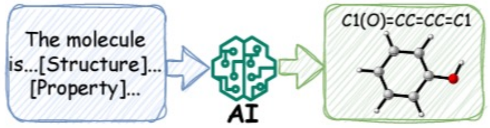
- **Molecule discovery**, etc.

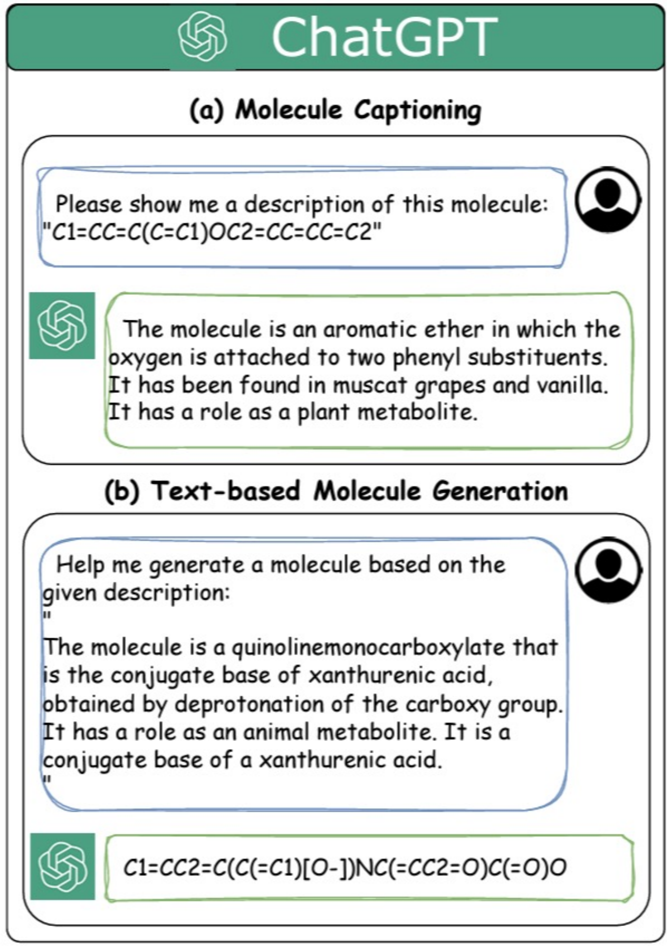


(a) Molecule Representations.



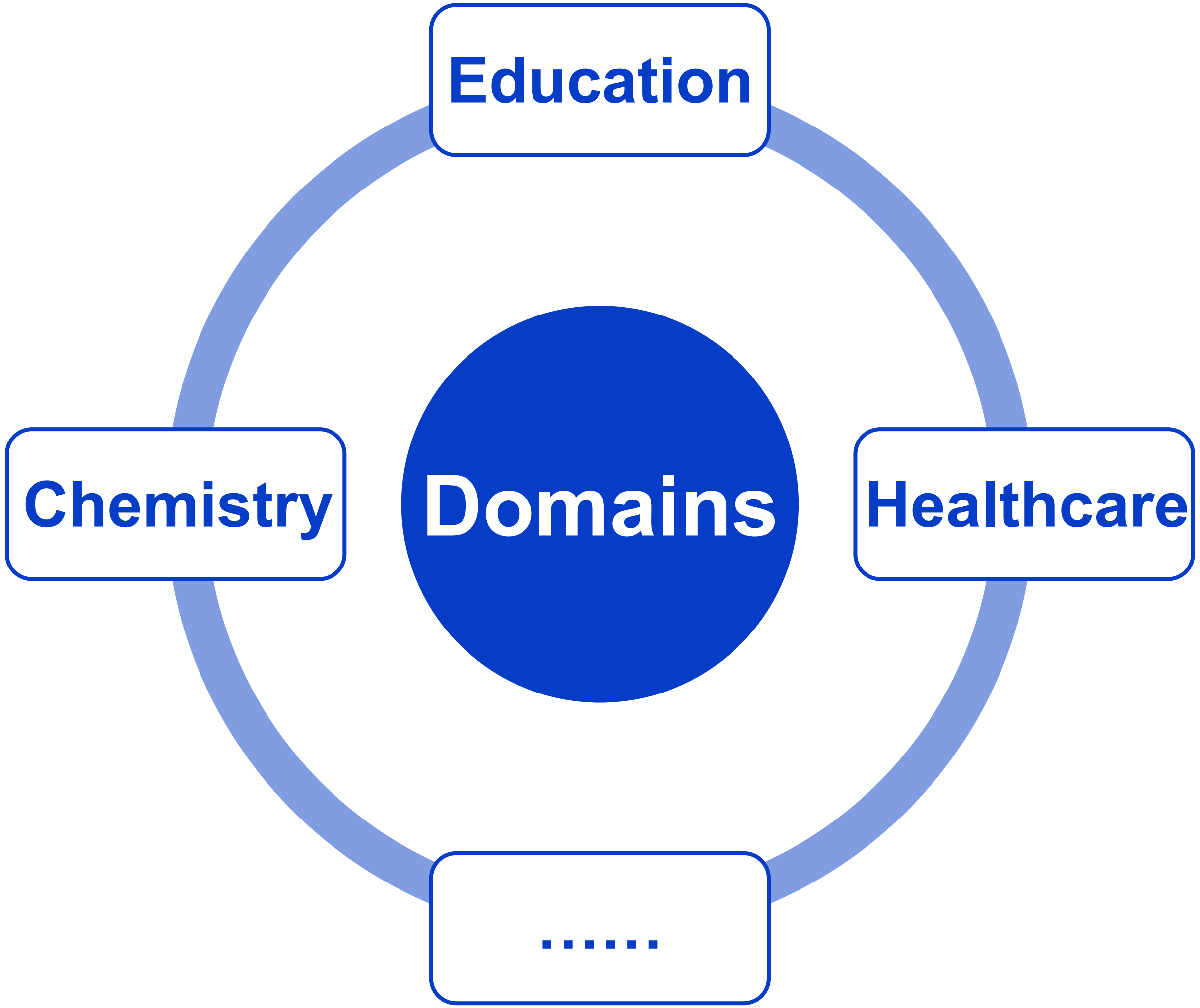
(b) Molecule Captioning.



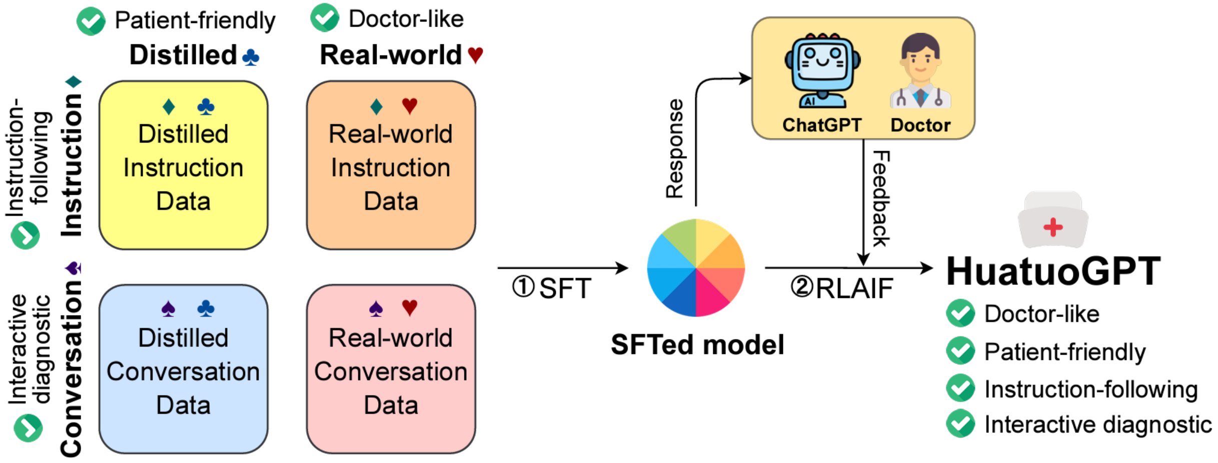


"Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective." TKDE, 2024. arXiv preprint arXiv:2306.06615.
"MolecularGPT: Open Large Language Model (LLM) for Few-Shot Molecular Property Prediction." arXiv preprint arXiv:2406.12950

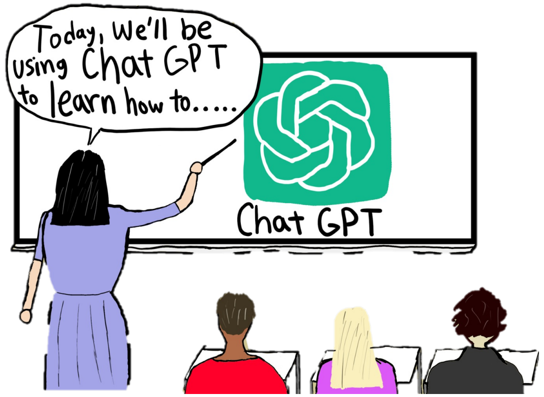
LLMs in Downstream Domains



❑ **Medical consultation**, etc.

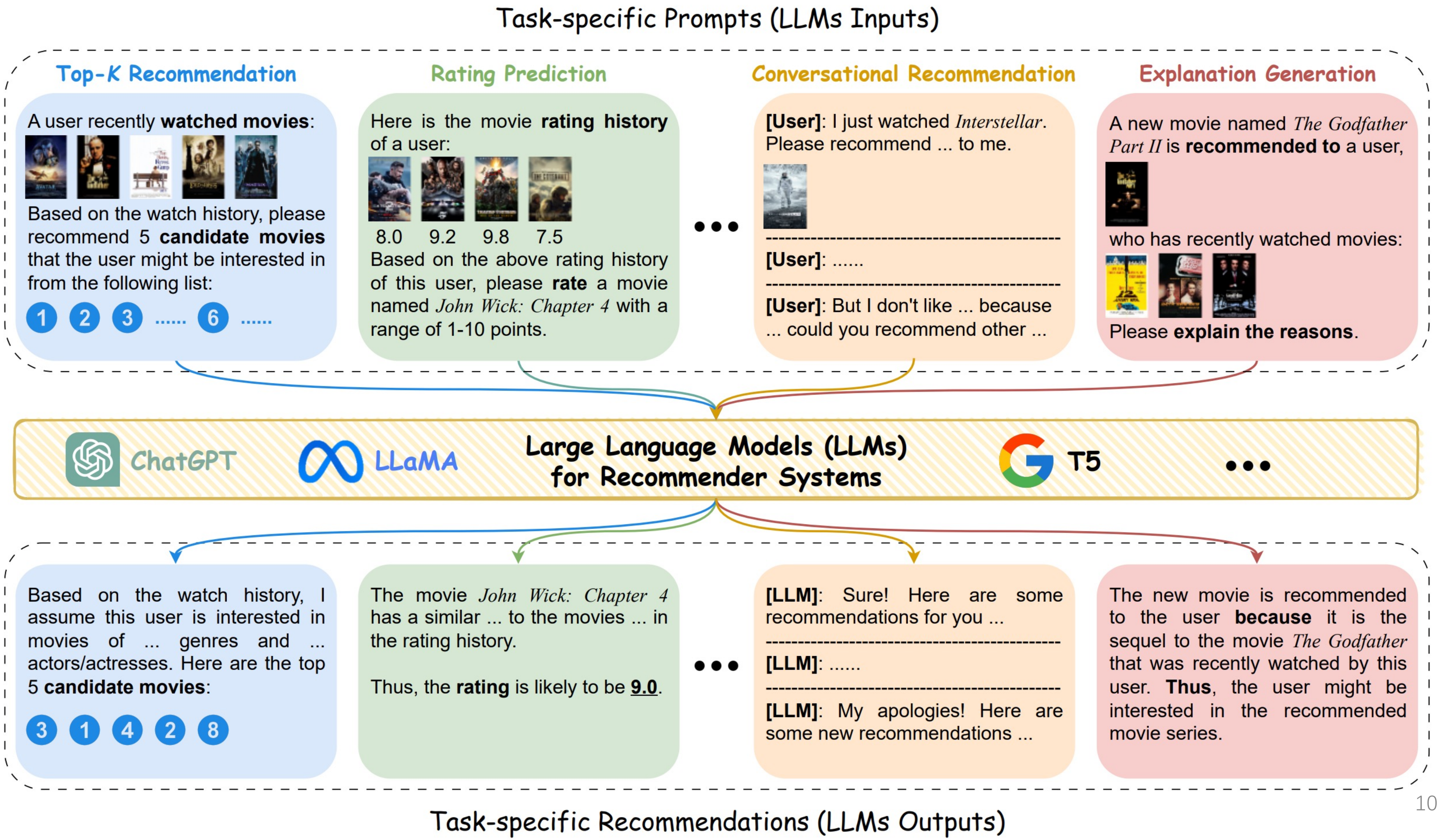


❑ **Curriculum & Teaching**, etc.



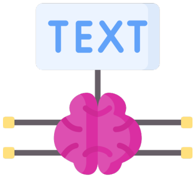
"HuatuoGPT, towards Taming Language Model to Be a Doctor." arXiv preprint arXiv:2305.15075 (2023).

LLMs in RecSys

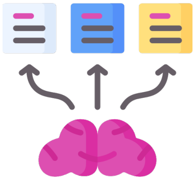


Potentials of LLMs in RecSys

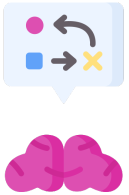
As the **parameter size** of LLMs continues to **scale up** with a larger **training corpus ...**



- ❑ Language understanding and generation ability
  - ❖ LLMs can comprehend **human intentions** and generate **language responses** that are more human-like in nature.



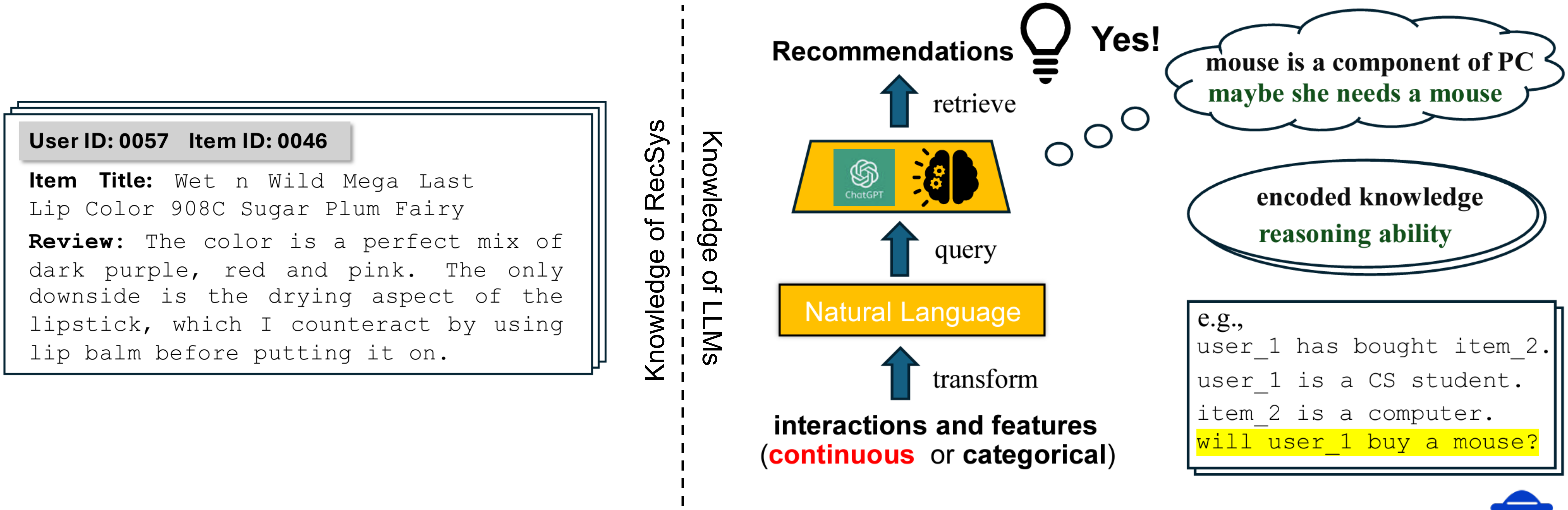
- ❑ Generalization capability
  - ❖ LLMs can apply their learned knowledge to **fit various downstream tasks**, even **without being fine-tuned** on specific tasks.



- ❑ Reasoning capability
  - ❖ LLMs can generate the outputs with **step-by-step reasonings** to support complex **decision-making processes**.

Language Understanding & Generation

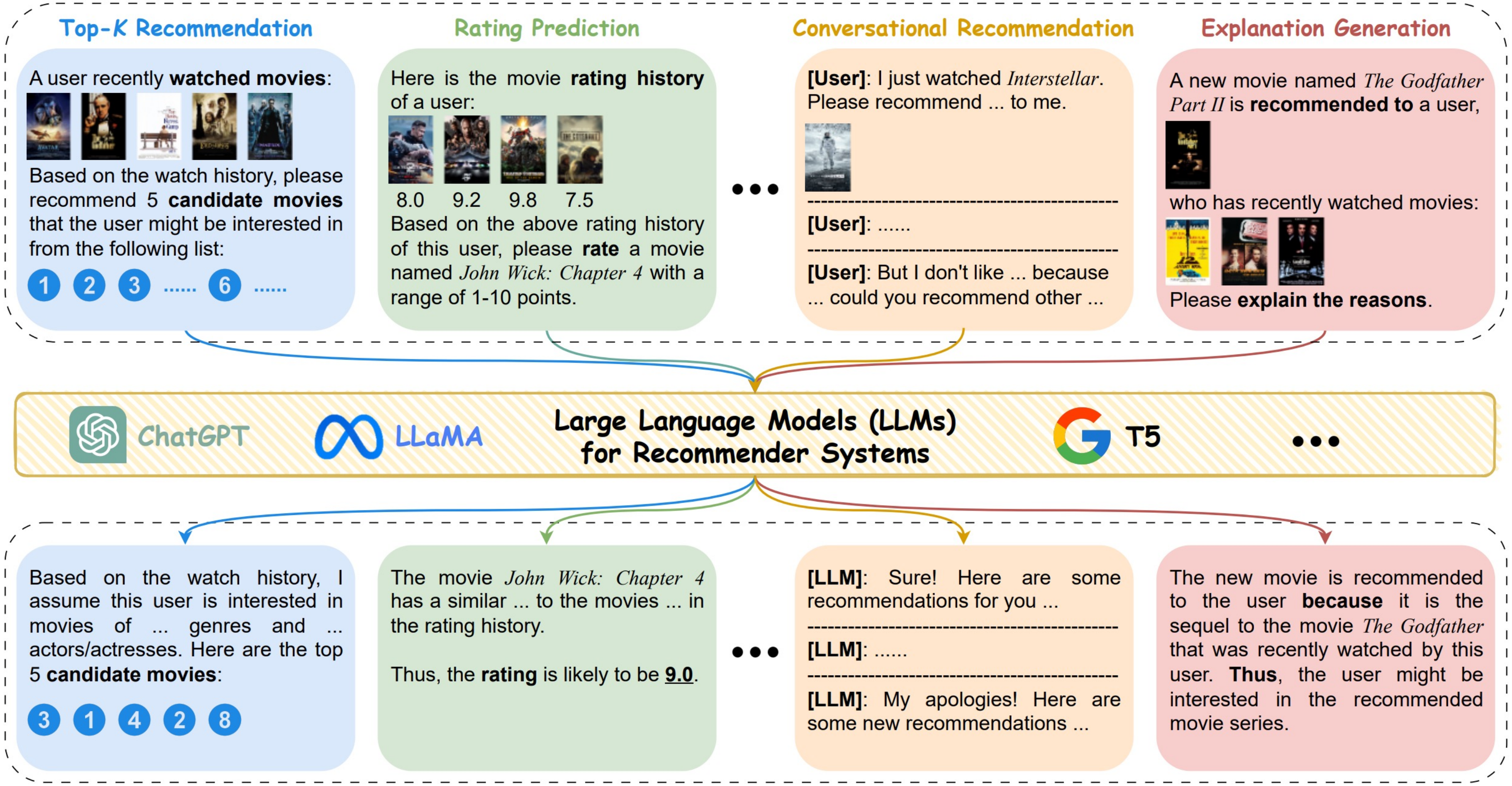
- Sufficiently capture **textual knowledge** about users and items
  - Rich **textual side information** about users and items in RecSys
  - Diverse **open-world knowledge** encoded in LLMs



"Collaborative Large Language Model for Recommender Systems." arXiv preprint arXiv:2311.01343 (2023).

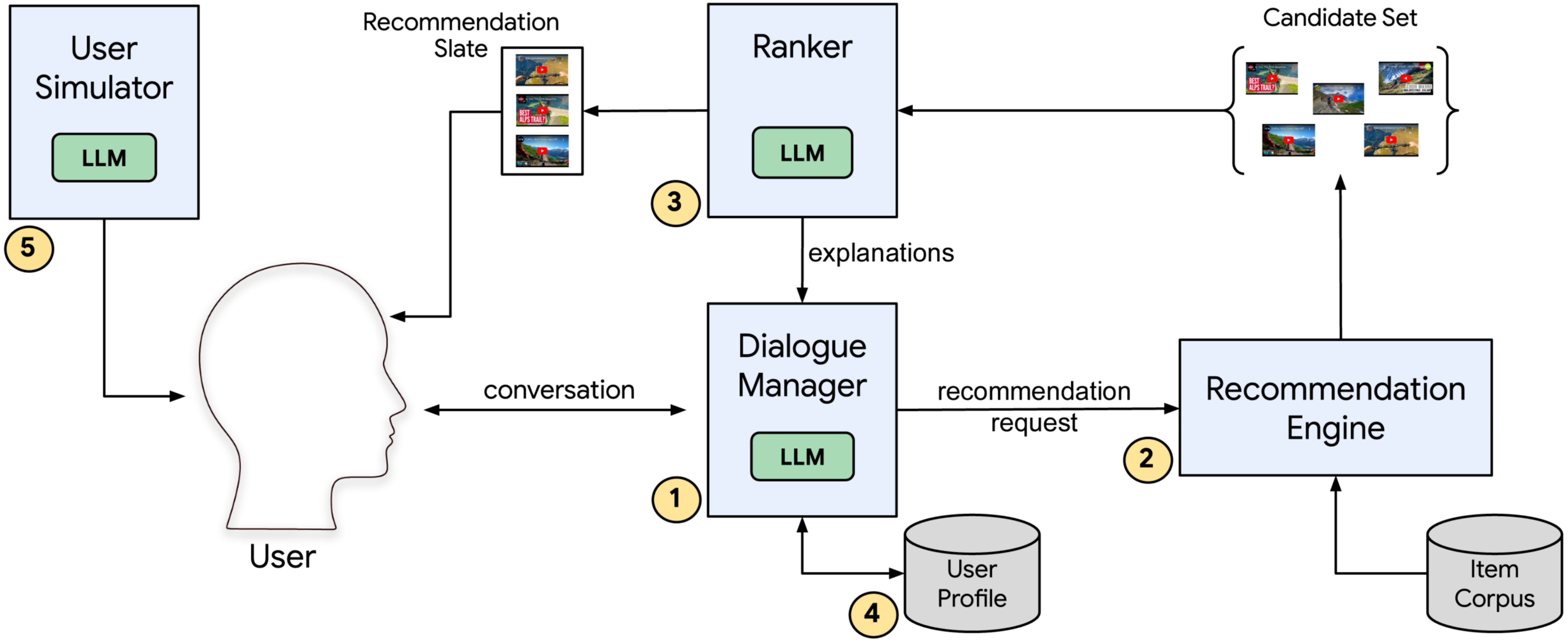
Generalization

- Adapt to **various recommendation tasks** even without being fine-tuned
  - LLMs can apply their **learned knowledge** to address recommendation objectives
  - **Multi-task adaption** by providing appropriate task instructions or a few task demonstrations



Reasoning

- Support complex **decision-making processes** in RecSys
  - Retrieve information from **large contexts** and control **multi-step** recommendation tasks
  - Generate outputs with **step-by-step reasoning** empowered by chain-of-thought prompting



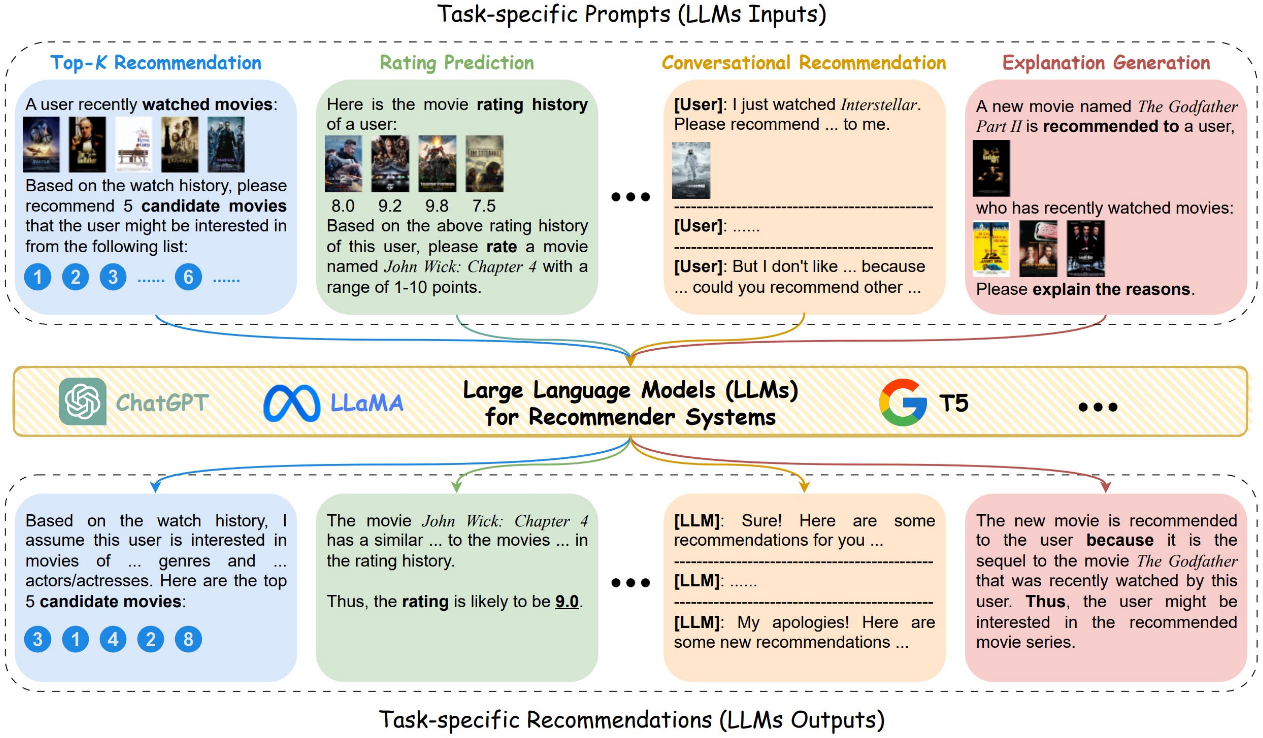
"Leveraging Large Language Models in Conversational Recommender Systems." arXiv preprint arXiv:2305.07961 (2023).

A Comprehensive Survey Paper

Recommender Systems in the Era of Large Language Models (LLMs)

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li

https://arxiv.org/abs/2307.02046



Survey

Zhao Z, Fan W, Li J, et al. Recommender systems in the era of large language models (llms)[J]. TKDE, 2024.

Recruitment

- Our research group are actively recruiting self-motivated **postdoc, Ph.D. students, and research assistants**, etc. **visiting scholars, interns, and self-funded students** are also welcome. Send me an email if you are interested.

  - Research areas: machine learning (ML), data mining (DM), artificial intelligence (AI), deep learning (DNNs), large language models (LLMs), graph neural networks (GNNs), computer vision (CV), natural language processing (NLP), etc.

  - Position details: https://wenqifan03.github.io/openings.html

Tutorial Outline

- **Part 1: Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2: Preliminaries of RecSys and LLMs (Dr. Yujuan Ding)**
- **Part 3: Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4: Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5: Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 5: Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

PART 2: Preliminaries of RecSys and LLMs

Presenter
Dr. Yujuan DING
HK PolyU

- **Recommender Systems (RecSys)**
  - Collaborative Filtering (CF)
  - Content-based Recommendation
  - Deep Recommender Systems
- **Large Language Models (LLMs)**
  - Development and Capability
  - LLM Architecture
- **LLM-based RecSys**
  - ID-based LLM RecSys
  - Text-based LLM RecSys

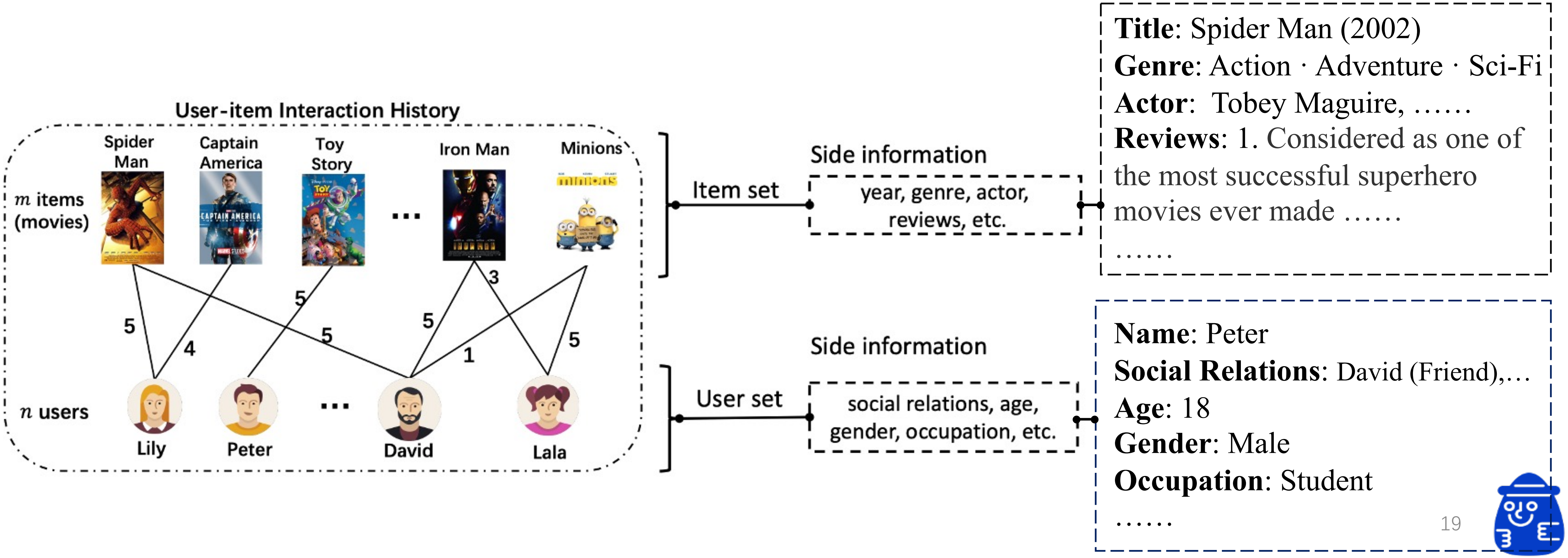
Recommender Systems

INPUT

Historical user-item interactions or additional side information (e.g., social relations, item attributes, etc)

OUTPUT

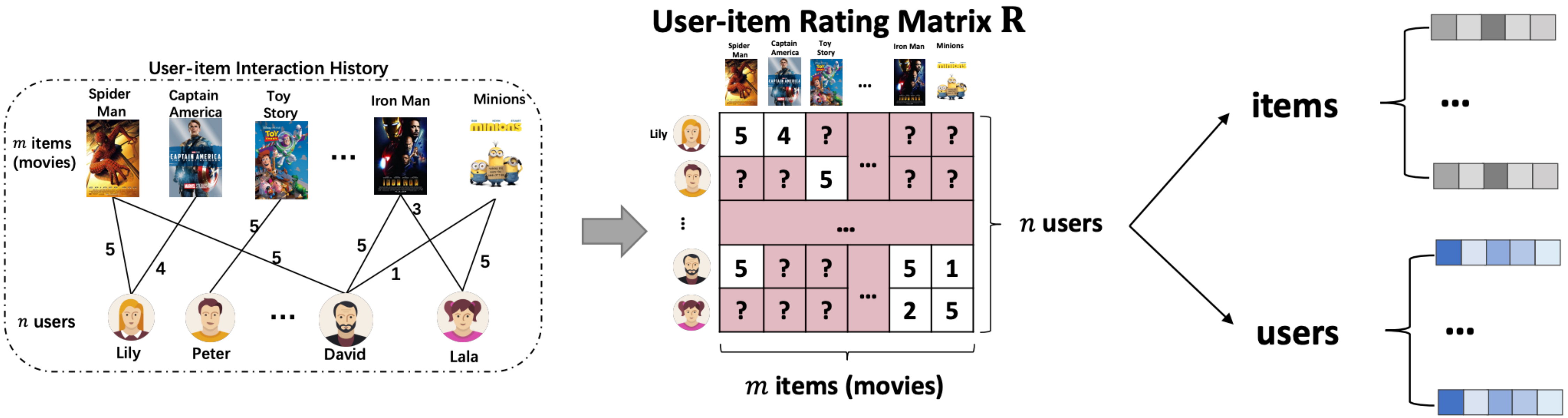
Predictions on how likely a user would be interested or interact (click, view, purchase, etc) with a target item



Collaborative Filtering (CF)-based Recommendation

- **CF** for recommendation
  - Similar users (with respect to their historical interactions) have similar preferences
  - Modelling user's preferences on items based on their past interactions (e.g., ratings and clicks)

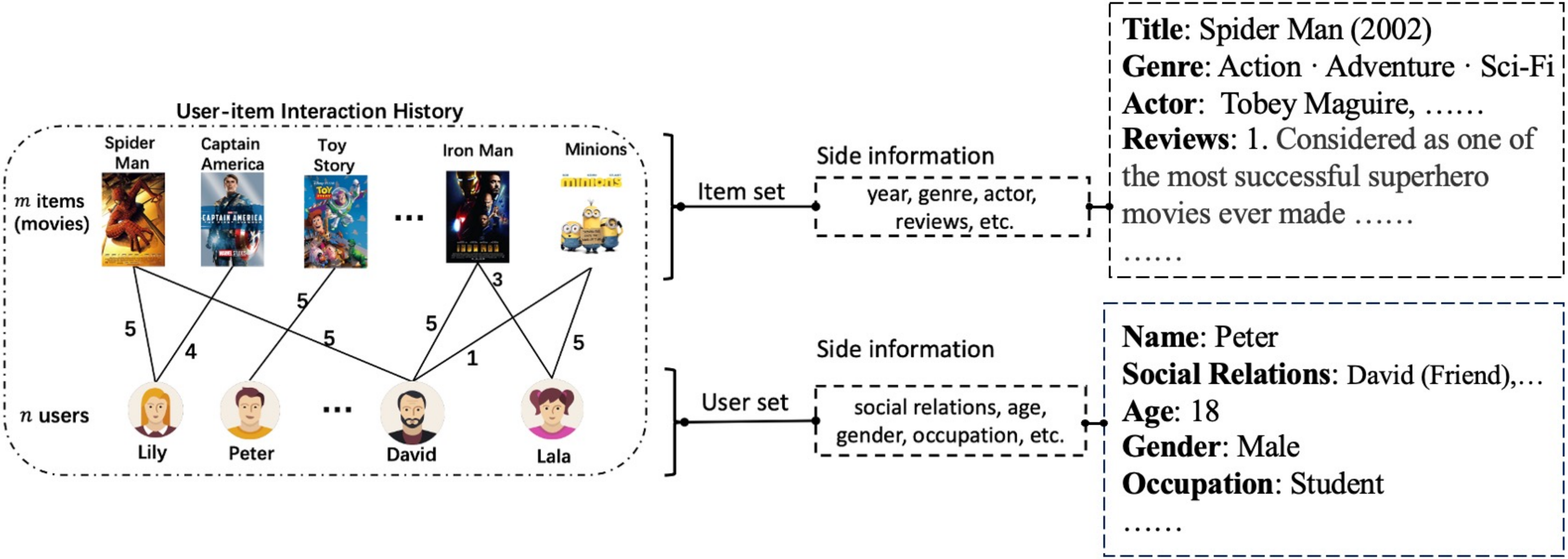
- **Learning representations** of users and items is the key to CF



Task: predicting missing movie ratings in Netflix.

Content-based Recommendation

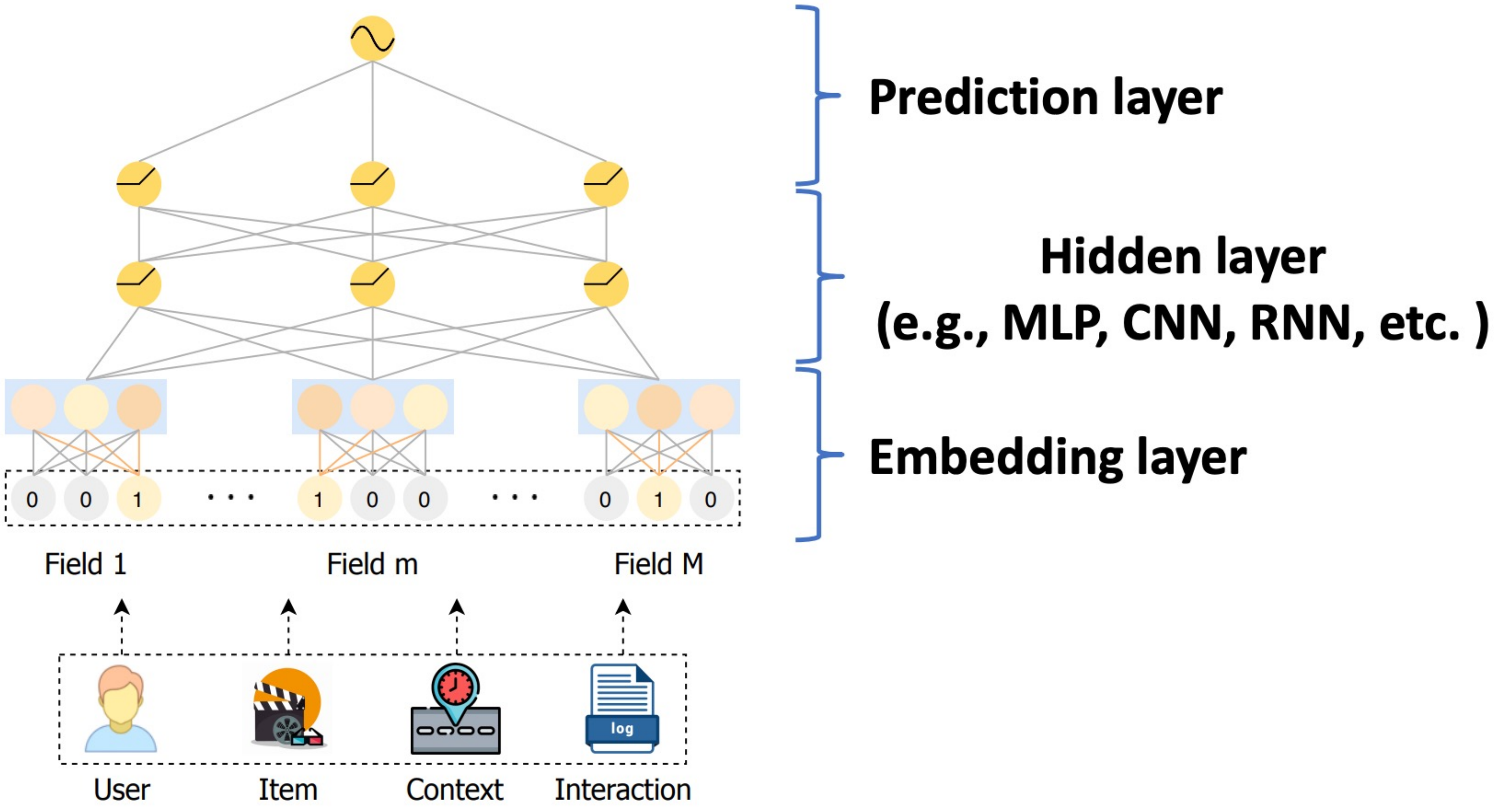
- Taking advantage of additional knowledge/information about users or items
- Enhancing user and item representations for improving recommendation performance



- Collaborative filtering + content == hybrid recommendation

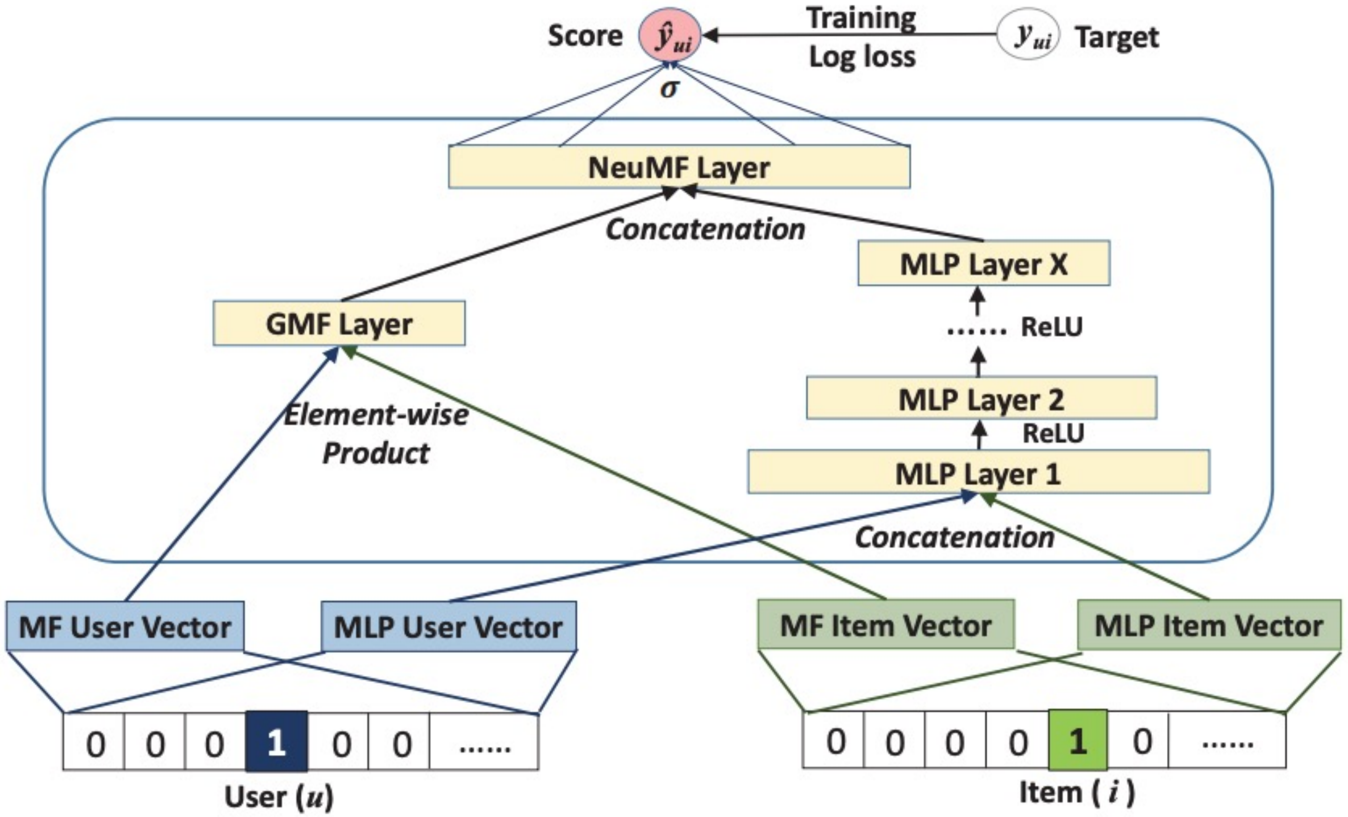
Deep Recommender Systems

- Deep learning techniques have been effectively applied to develop recommender systems
- Remarkable representation learning capabilities



NeuMF

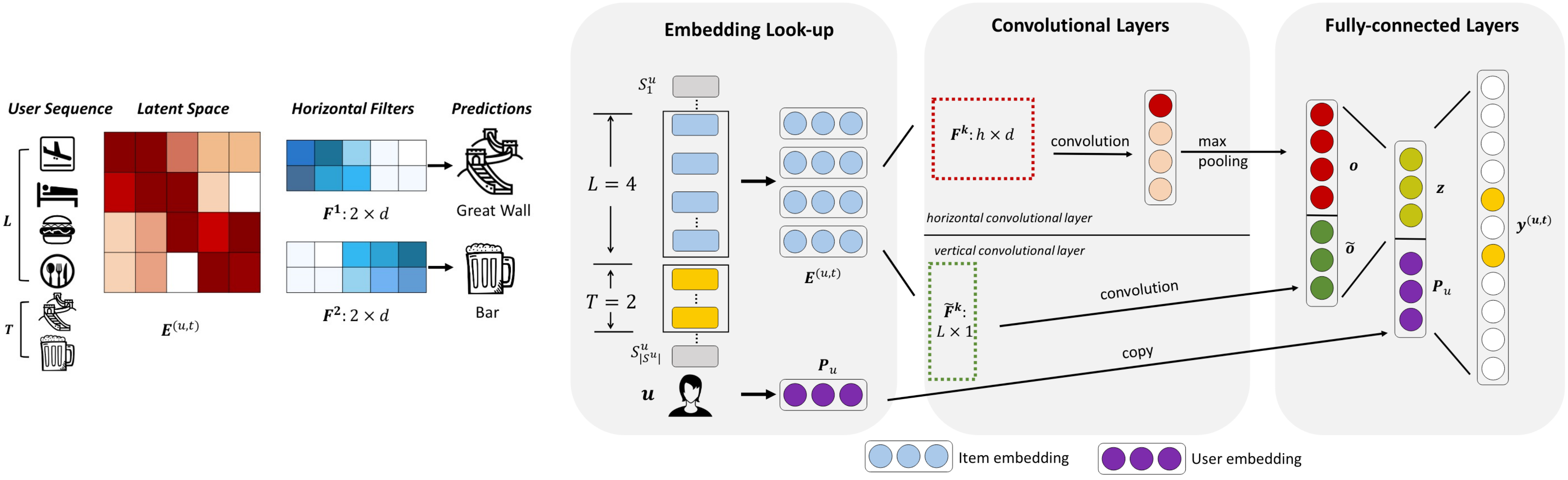
- **Neural Matrix Factorization (NeuMF)** unifies the strengths of MF and MLP in modelling user-item interactions
  - MF uses an inner product as the interaction function
  - MLP may be more capable to capture the complex structure of the interaction patterns



"Neural collaborative filtering." Proceedings of the 26th international conference on world wide web (2017).

Caser

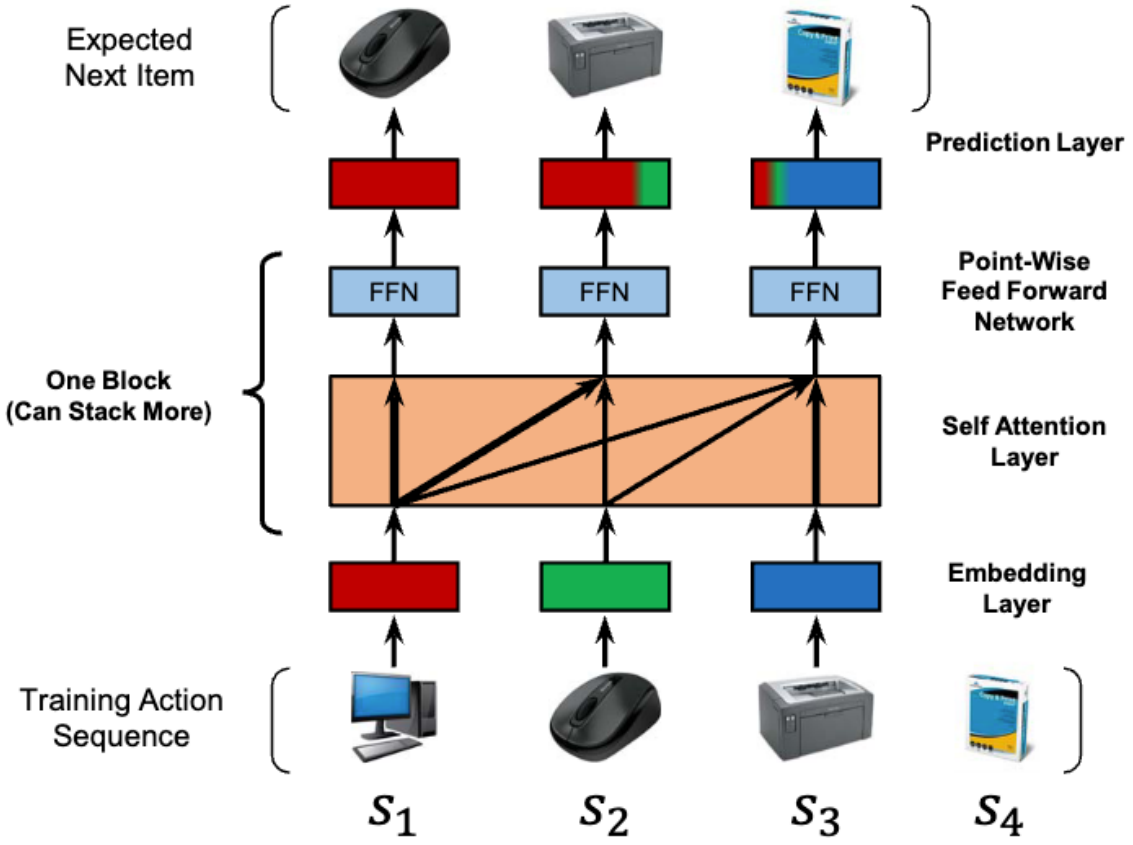
- Top-N sequential recommendation models each user as a sequence of items interacted in the past and aims to predict top-N ranked items
- Convolutional Sequence Embedding Recommendation Model



"Personalized top-n sequential recommendation via convolutional sequence embedding." WSDM (2018).

SASRec

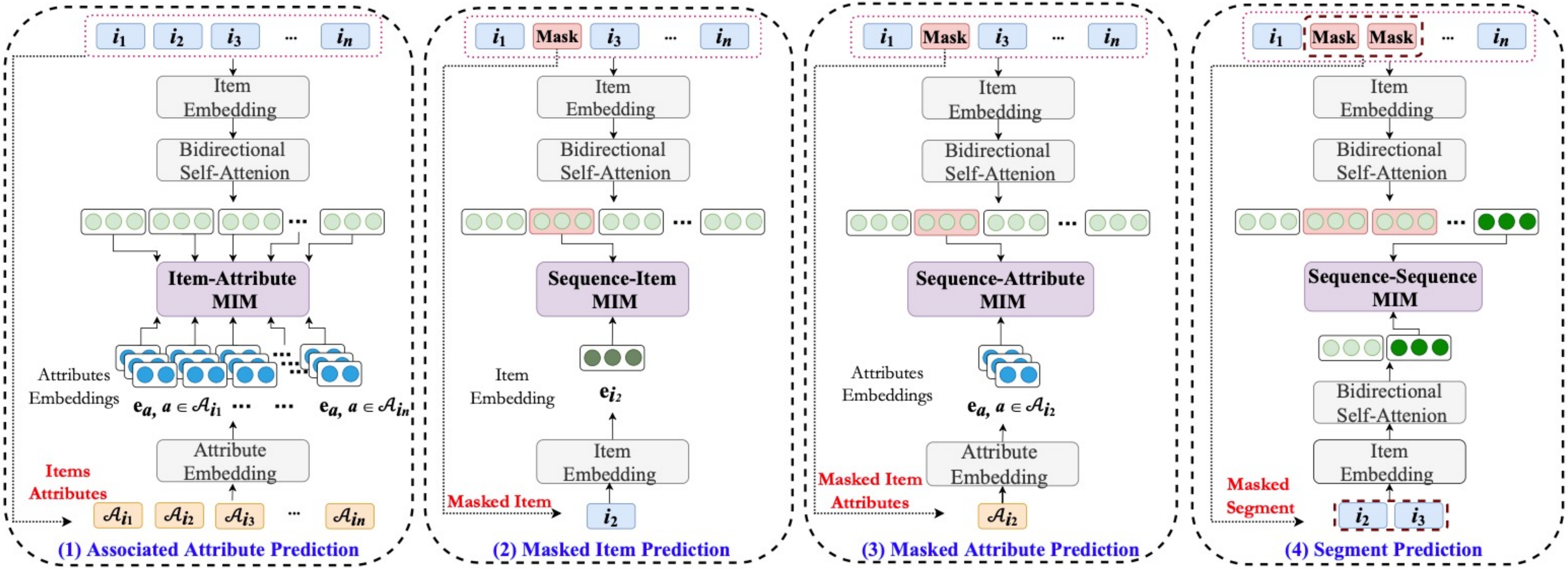
- Self-Attentive Sequential Recommendation
- Using an **attention** mechanism to capture **long-term semantics** and makes its **predictions** based on relatively **few actions**



"Self-attentive sequential recommendation." ICDM (2018).

S3-Rec

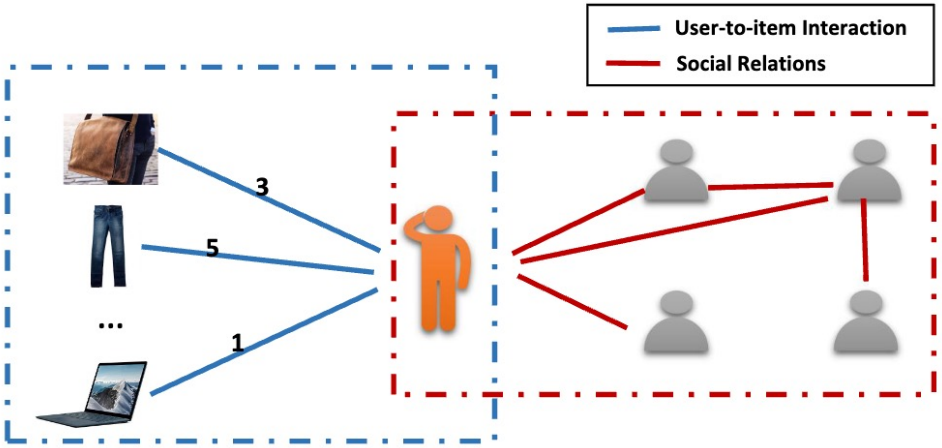
- Utilizing the intrinsic data correlation to derive **self-supervision** signals
- Enhancing the data representations via **pre-training** methods



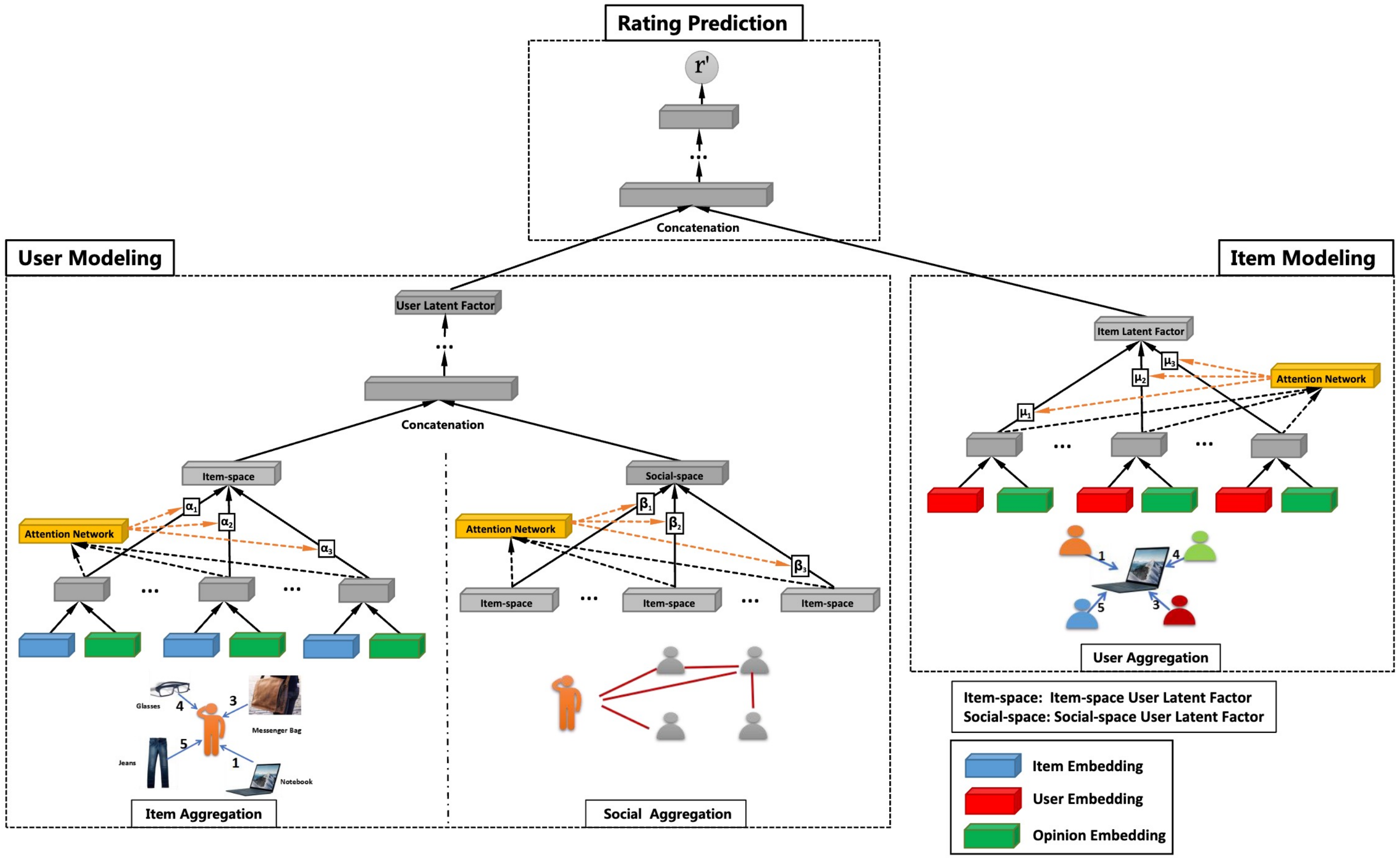
"S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization." CIKM (2020).

GraphRec

- Data in social recommender systems can be represented as **user-user** social graph and **user-item** graph



user-item graph user-user graph



"Graph neural networks for social recommendation." WWW (2019).

PART 2: Preliminaries of RecSys and LLMs

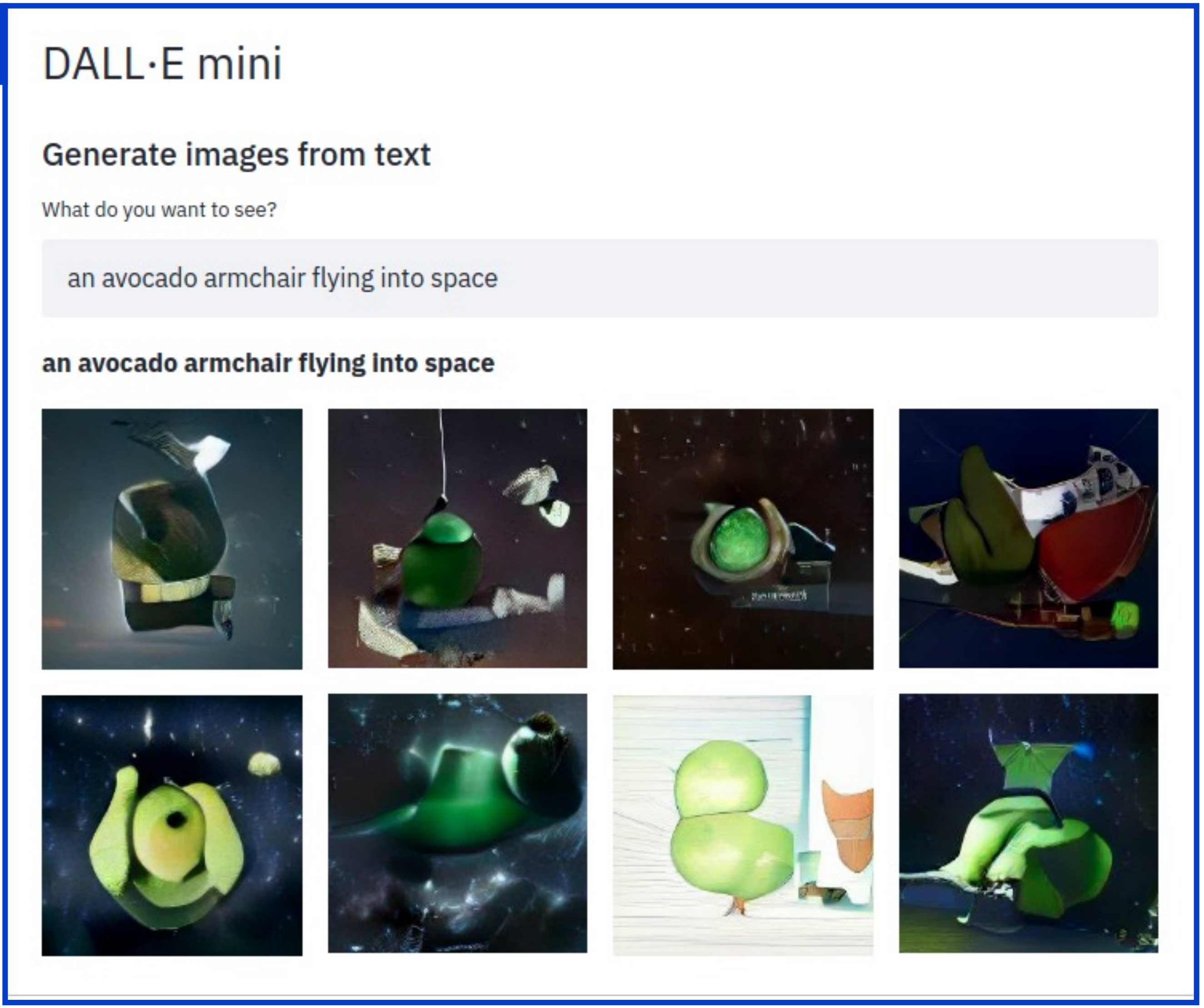
Website of this tutorial

- **Recommender Systems (RecSys)**
  - Collaborative Filtering (CF)
  - Content-based Recommendation
  - Deep Recommender Systems
- **Large Language Models (LLMs)**
  - Development and Capability
  - LLM Architecture
- **LLM-based RecSys**
  - ID-based LLM RecSys
  - Text-based LLM RecSys

Emergence of Large Language Models (LLMs)

- LLMs can be used for a variety of tasks, such as **Image Generation**

Text to Image

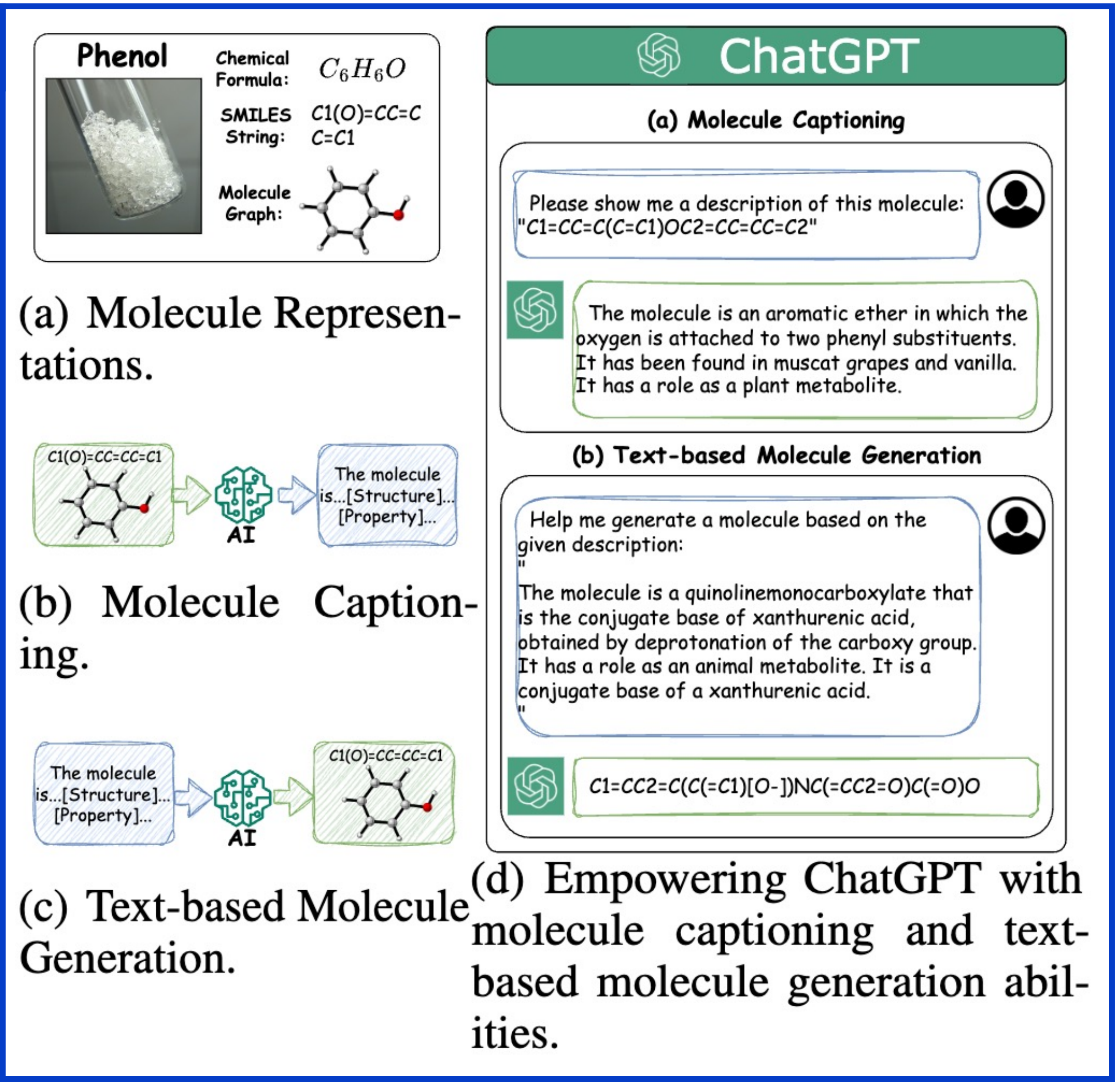


"Zero-shot text-to-image generation." *International Conference on Machine Learning*. PMLR, 2021.

Emergence of Large Language Models (LLMs)

- LLMs can be used for a variety of tasks, such as **Molecule Generation**

Text to Molecule



(a) Molecule Representations.

(b) Molecule Captioning.

(c) Text-based Molecule Generation.

(d) Empowering ChatGPT with molecule captioning and text-based molecule generation abilities.

"Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective." *arXiv* (2023).

Emergence of Large Language Models (LLMs)

- LLMs can be used for a variety of tasks, such as **Recommendation**

Text to Recommendation

Rating Prediction

zero-shot

How will user rate this product_title: "SHANY Nail Art Set (24 Famous Colors Nail Art Polish, Nail Art Decoration)" , and product_category: Beauty? (1 being lowest and 5 being highest) Attention! Just give me back the exact number a result , and you don't need a lot of text.

few-shot

Here is user rating history:
1. Bundle Monster 100 PC 3D Designs Nail Art Nailart Manicure Fimo Canes Sticks Rods Stickers Gel Tips, 5.0;
2. Winstonia's Double Ended Nail Art Marbling Dotting Tool Pen Set w/ 10 Different Sizes 5 Colors - Manicure Pedicure, 5.0;
3. Nail Art Jumbo Stamp Stamping Manicure Image Plate 2 Tropical Holiday by Cheeky&reg, 5.0 ;
4.Nail Art Jumbo Stamp Stamping Manicure Image Plate 6 Happy Holidays by Cheeky&reg, 5.0;
Based on above rating history, please predict user's rating for the product: "SHANY Nail Art Set (24 Famouse Colors Nail Art Polish, Nail Art Decoration)", (1 being lowest and5 being highest,The output should be like: (x stars, xx%), do not explain the reason.)

Sequential Recommendation

zero-shot

Requirements: you must choose 10 items for recommendation and sort them in order of priority, from highest to lowest. Output format: a python list. Do not explain the reason or include any other words.
The user has interacted with the following items in chronological order: ['Better Living Classic Two Chamber Dispenser, White', 'Andre Silhouettes Shampoo Cape, Metallic Black', ……, 'John Frieda JFHA5 Hot Air Brush, 1.5 inch'].Please recommend the next item that the user might interact with.

few-shot

Requirements: you must choose 10 items for recommendation and sort them in order of priority, from highest to lowest. Output format: a python list. Do not explain the reason or include any other words.
Given the user's interaction history in chronological order: ['Avalon Biotin B-Complex Thickening Conditioner, 14 Ounce', 'Conair 1600 Watt Folding Handle Hair Dryer', ……, 'RoC Multi-Correxion 4-Zone Daily Moisturizer, SPF 30, 1.7 Ounce'], the next interacted item is ['Le Edge Full Body Exfoliator - Pink']. Now, if the interaction history is updated to ['Avalon Biotin B-Complex Thickening Conditioner, 14 Ounce', 'Conair 1600 Watt Folding Handle Hair Dryer',……, 'RoC Multi-Correxion 4-Zone Daily Moisturizer, SPF 30, 1.7 Ounce', 'Le Edge Full Body Exfoliator - Pink'] and the user is likely to interact again, recommend the next item.

"Is ChatGPT a good recommender? a preliminary study." CIKM (2023).

What are Language Models?

- Narrow Sense
  - A **probabilistic model** that assigns a probability to every **finite sequence** (grammatical or not)

Sentence: "the cat sat on the mat"

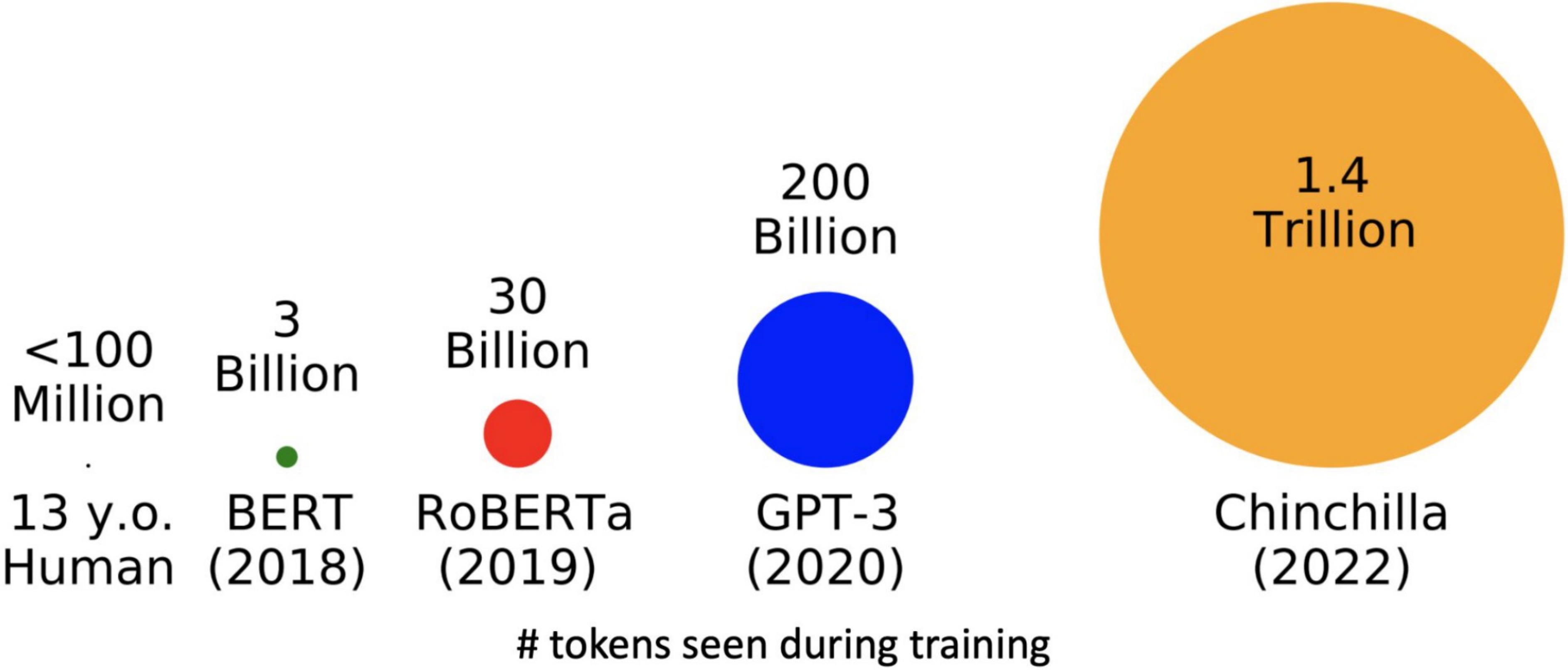
$$
\begin{aligned}
P(\text{the cat sat on the mat}) = & \; P(\text{the}) * P(\text{cat}|\text{the}) * P(\text{sat}|\text{the cat}) \\
& * P(\text{on}|\text{the cat sat}) * P(\text{the}|\text{the cat sat on}) \\
& * P(\text{mat}|\text{the cat sat on the})
\end{aligned}
$$

Implicit order

- Broad Sense
  - Encoder-only models (BERT, RoBERTa, ELECTRA)
  - Decoder-only models (GPT-X, OPT, LLaMa, PaLM)
  - Encoder-decoder models (T5, BART)

Large Language Model Development

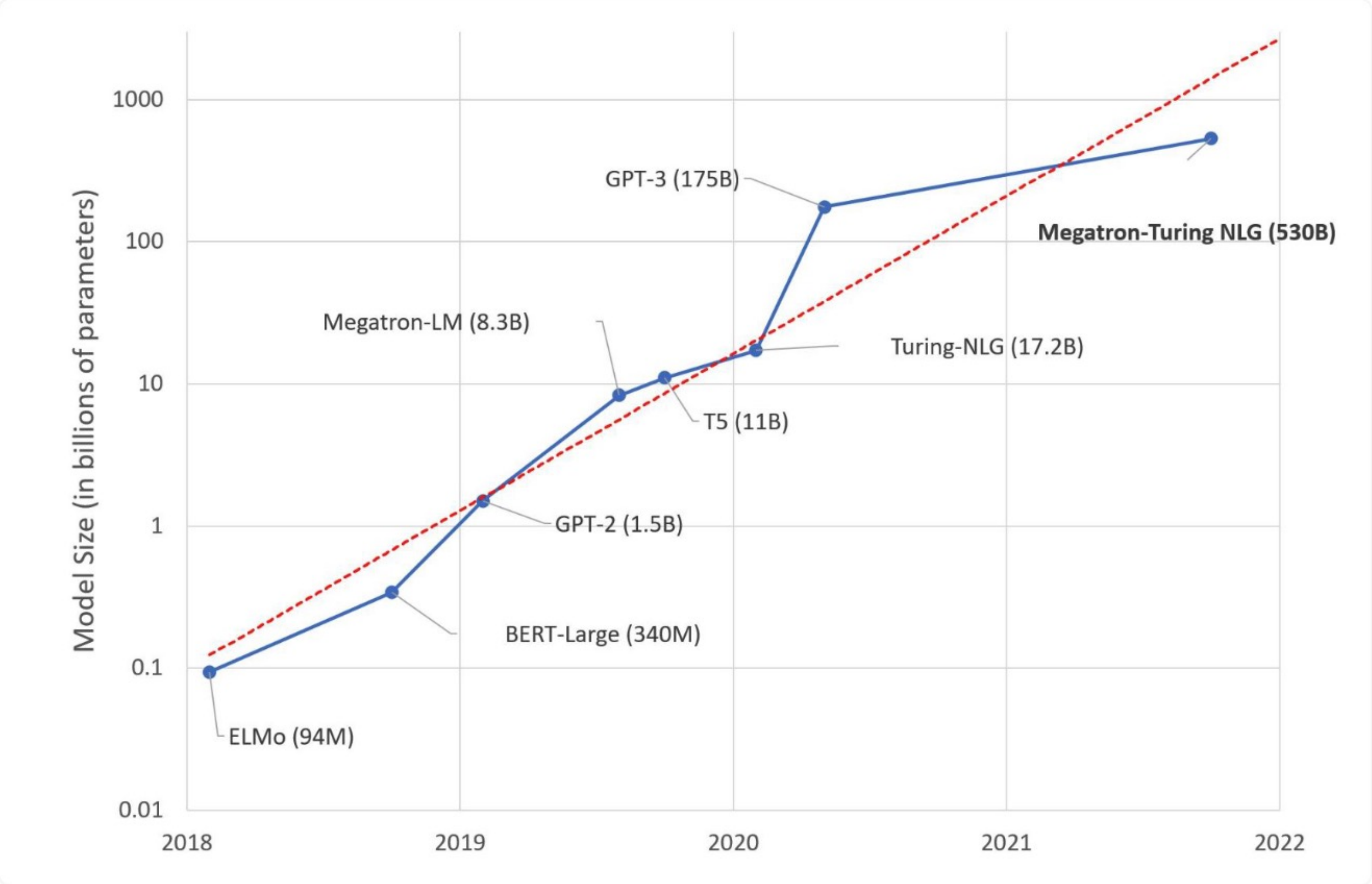
- ❑ Trained on more and more data – **Hundreds of Billions of Tokens**



https://babylm.github.io/

Large Language Model Development

- Larger and larger models – **Billions of Parameters**

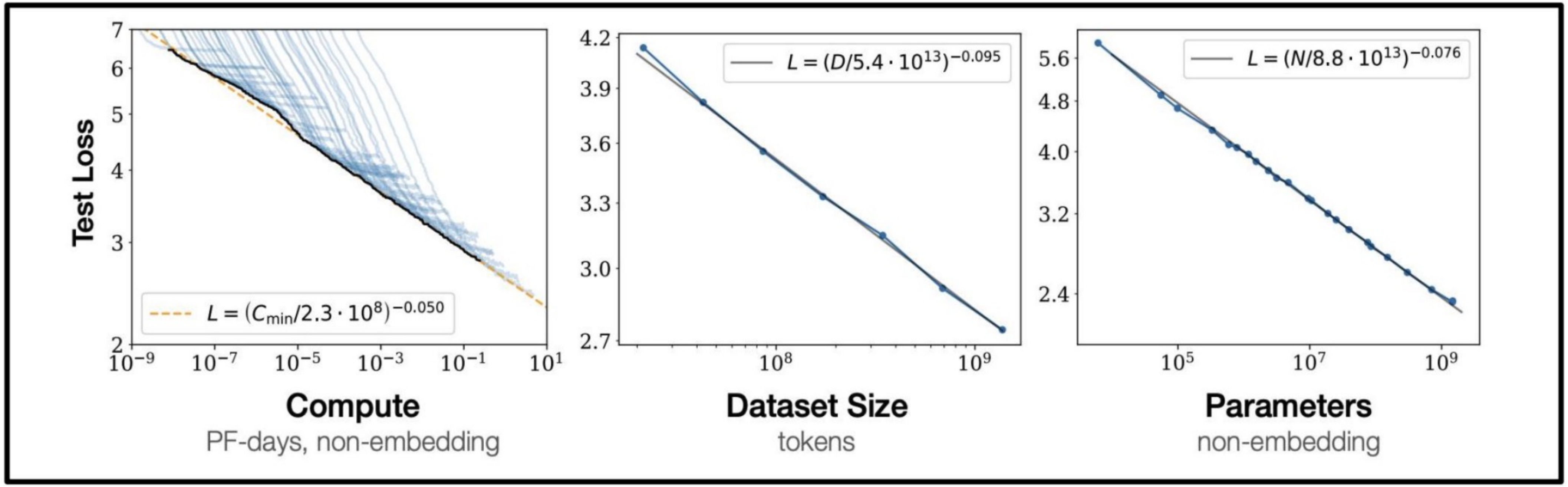


https://huggingface.co/blog/large-language-models

Why Large Language Models?

- Scaling Law for Neural Language Models

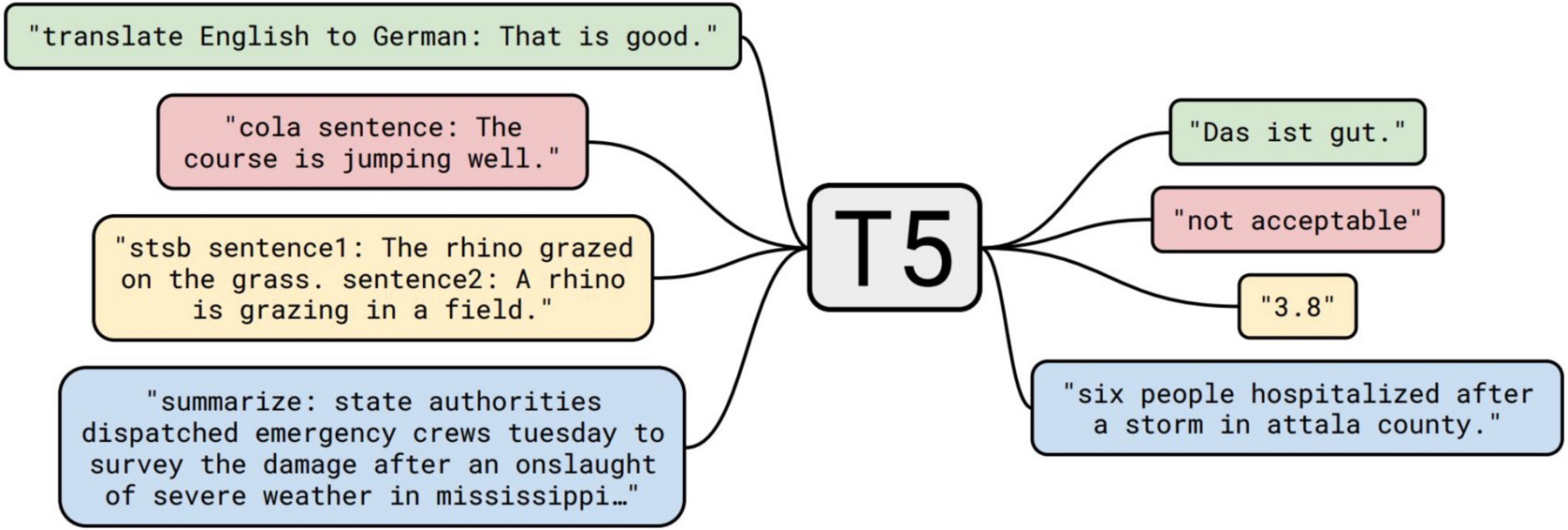
  - Performance depends strongly on scale! We keep getting better performance as we scale the model, data, and compute up!



"Scaling laws for neural language models." arXiv preprint (2020).

Why Large Language Models?

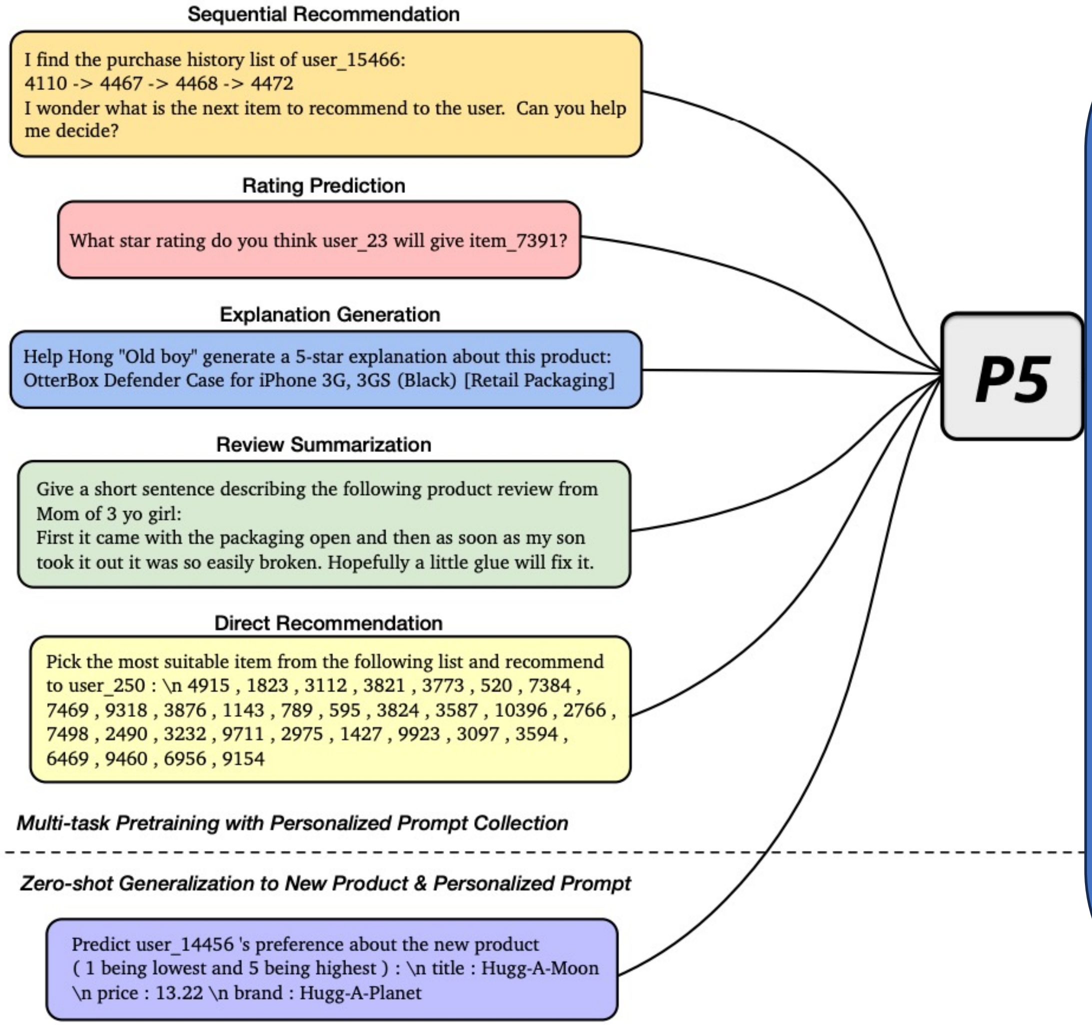
- Generalization
  - We can now use one single model to solve many NLP tasks.



"Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research (2020).

Why Large Language Models?

- Strong Zero-shot/Few-shot Ability



Multiple Tasks in One Model

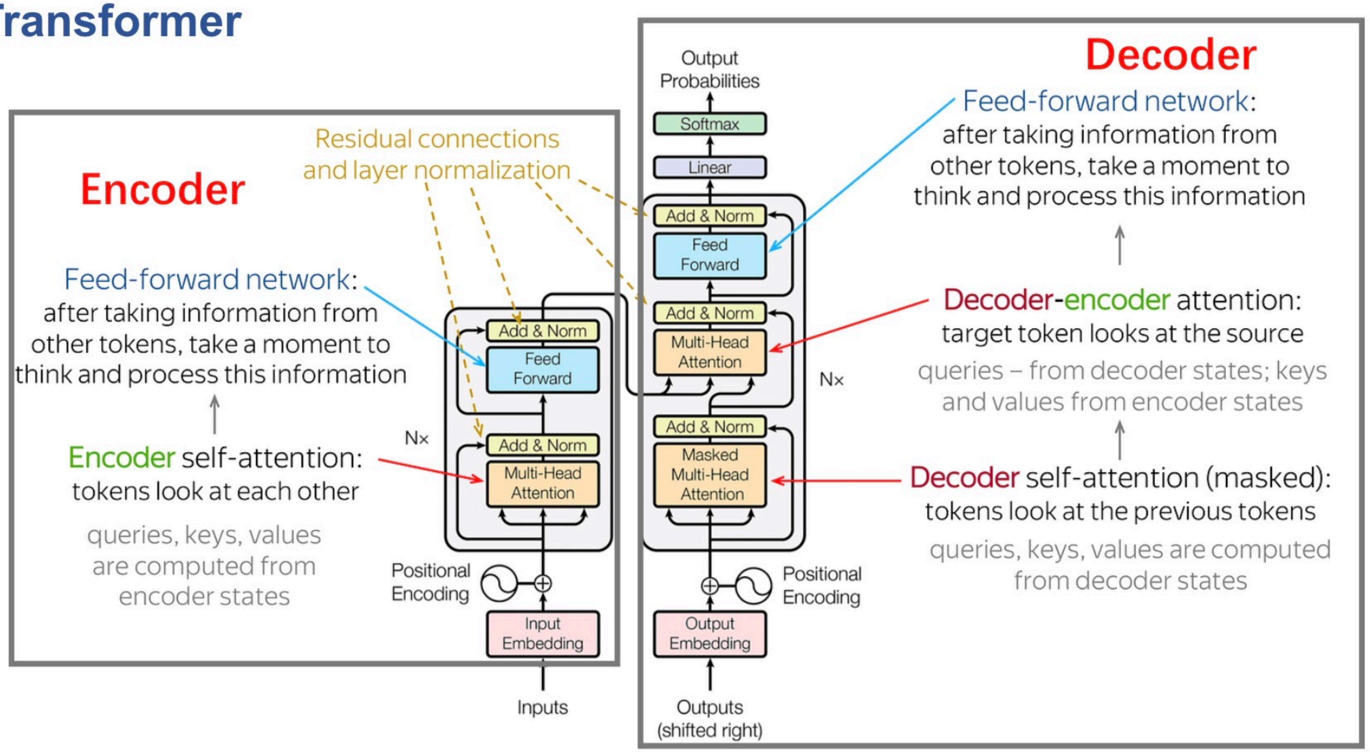
- Sequential recommendation
- Rating prediction
- Explain generation
- Review summarization
- Direct recommendation

"Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys (2022).

Large Language Model Structure

- Encoder-Only Models
  - BERT, RoBERTa, ELECTRA
- Decoder-Only Models
  - GPT-X, OPT, LLaMa, PaLM
- Encoder-Decoder Models
  - T5, BART

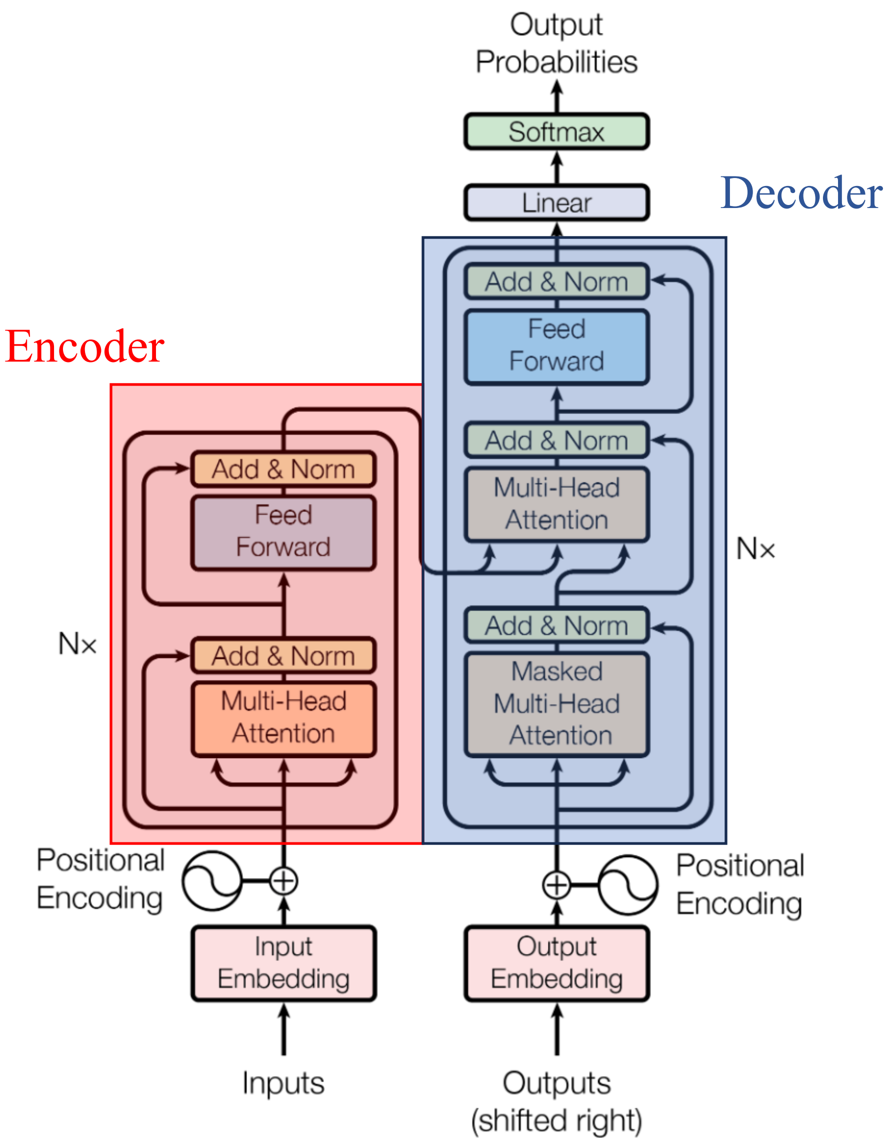
Transformer

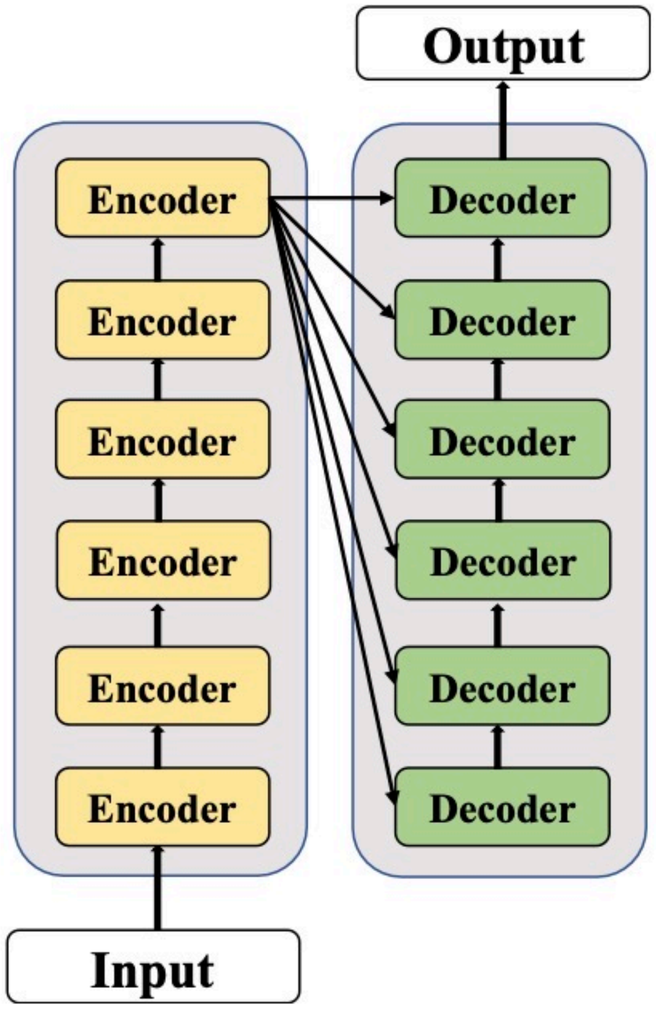


The Transformer – model architecture

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Transformer

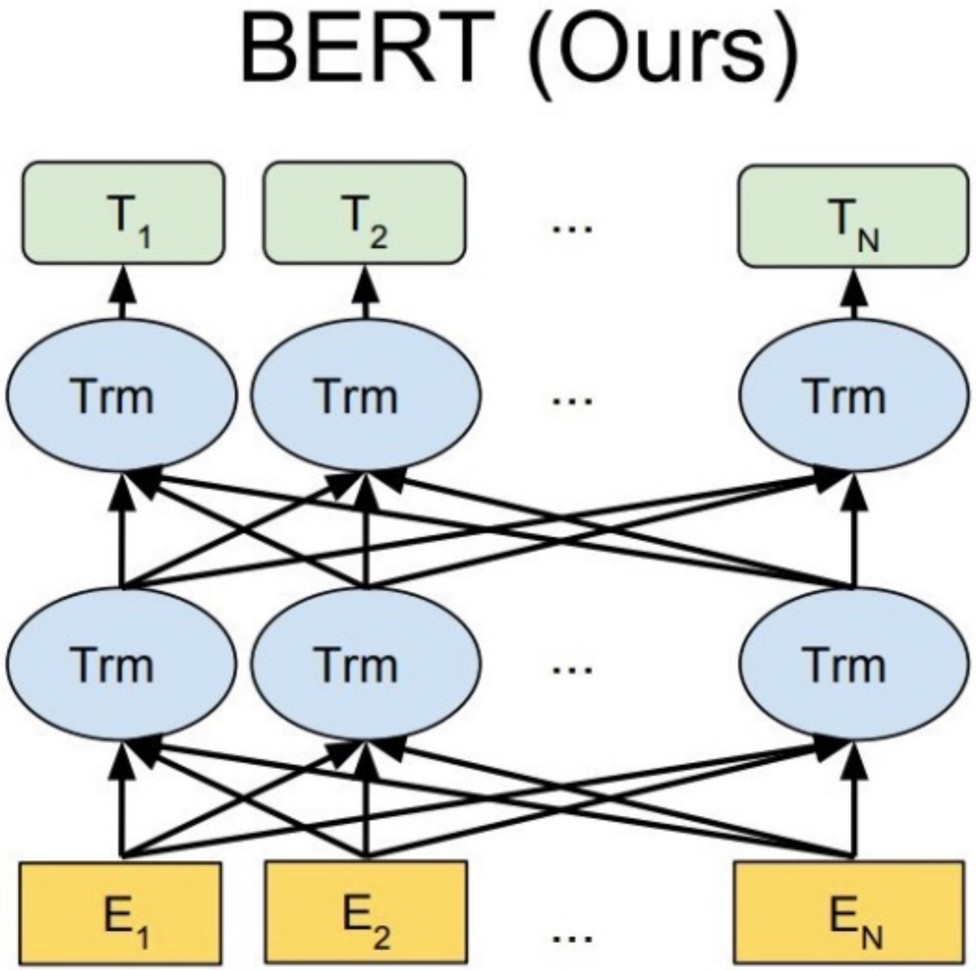




Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Encoder-Only Models: BERT

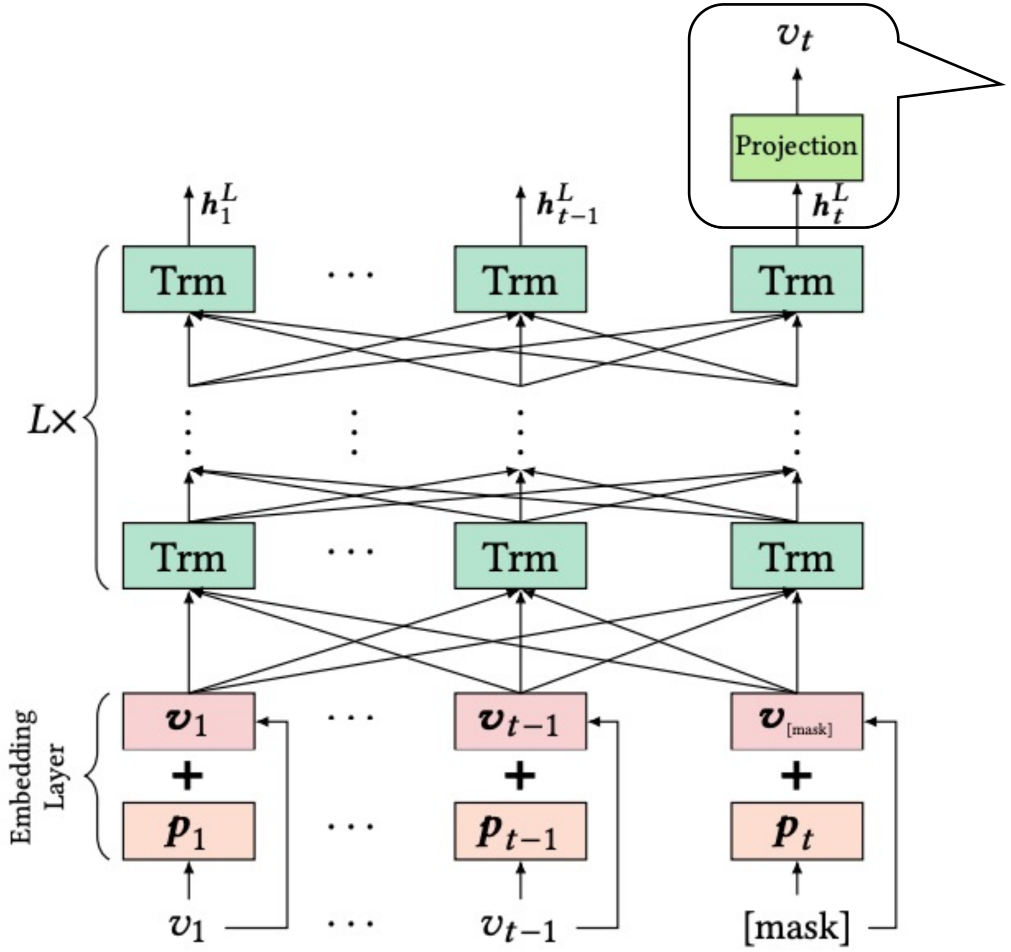
- BERT uses a **bidirectional** Transformer



"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint (2018).

Encoder-Only Models for Rec: BERT4Rec

- Adopt **Bidirectional Encoder Representations** from Transformers to model the sequential nature of user behaviors



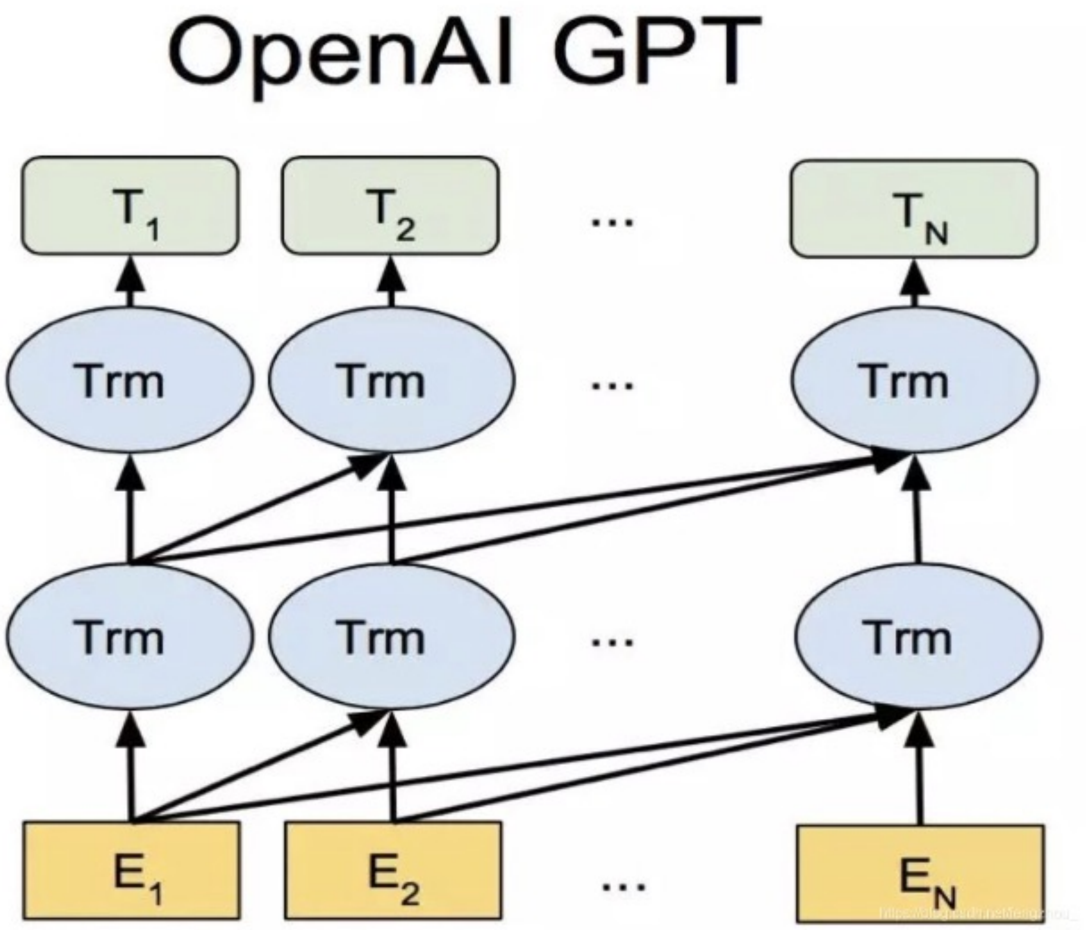
Next-item prediction

Natural Language		**Recommendation**
Token sequence	→	Item ID sequence
Inter-token correlations	→	Inter-item correlations

"BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." CIKM (2019).

Decoder-Only Models: GPT

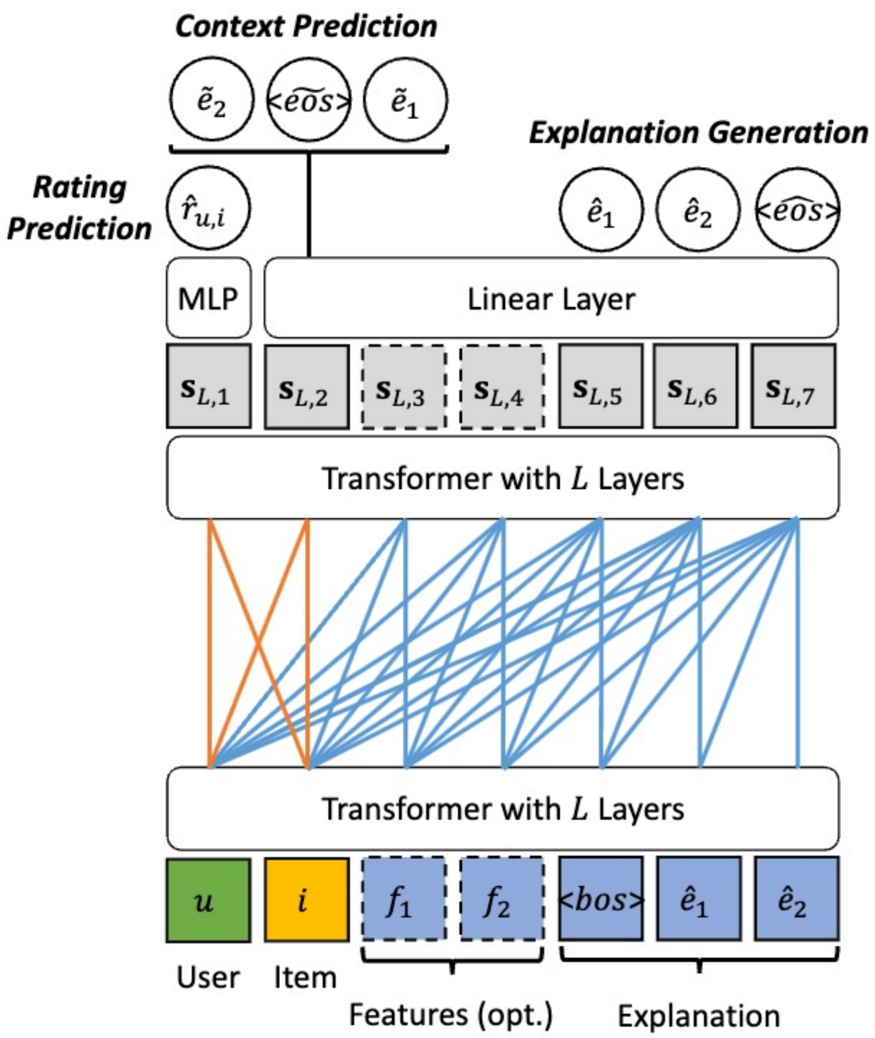
- OpenAI GPT uses a **left-to-right** Transformer



"Improving language understanding by generative pre-training." (2018).

Decoder-Only Models for Rec: PETER

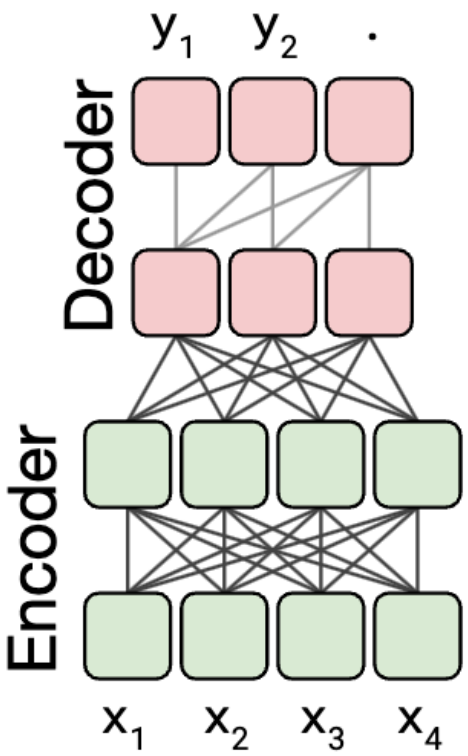
- Utilizing the IDs to predict the words in the target explanation

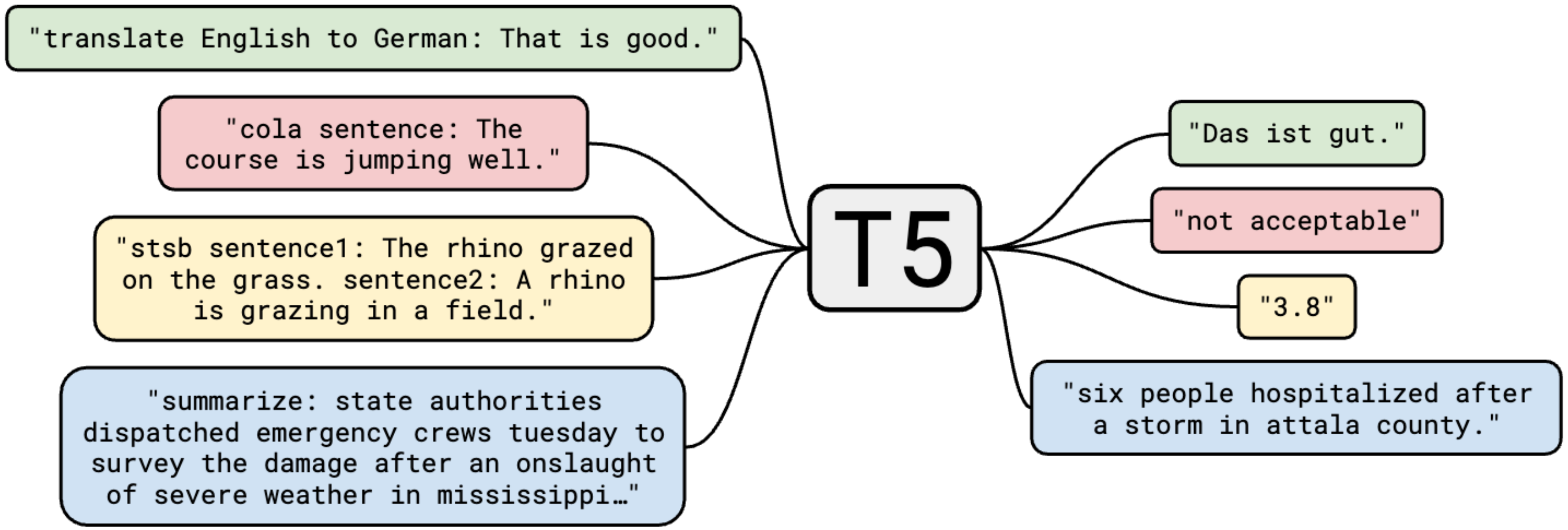


"Personalized transformer for explainable recommendation." ACL-IJCNLP (2021).

Encoder-Decoder Models: T5

- T5 handles any text-to-text task by converting every natural language processing problem into a text generation problem.

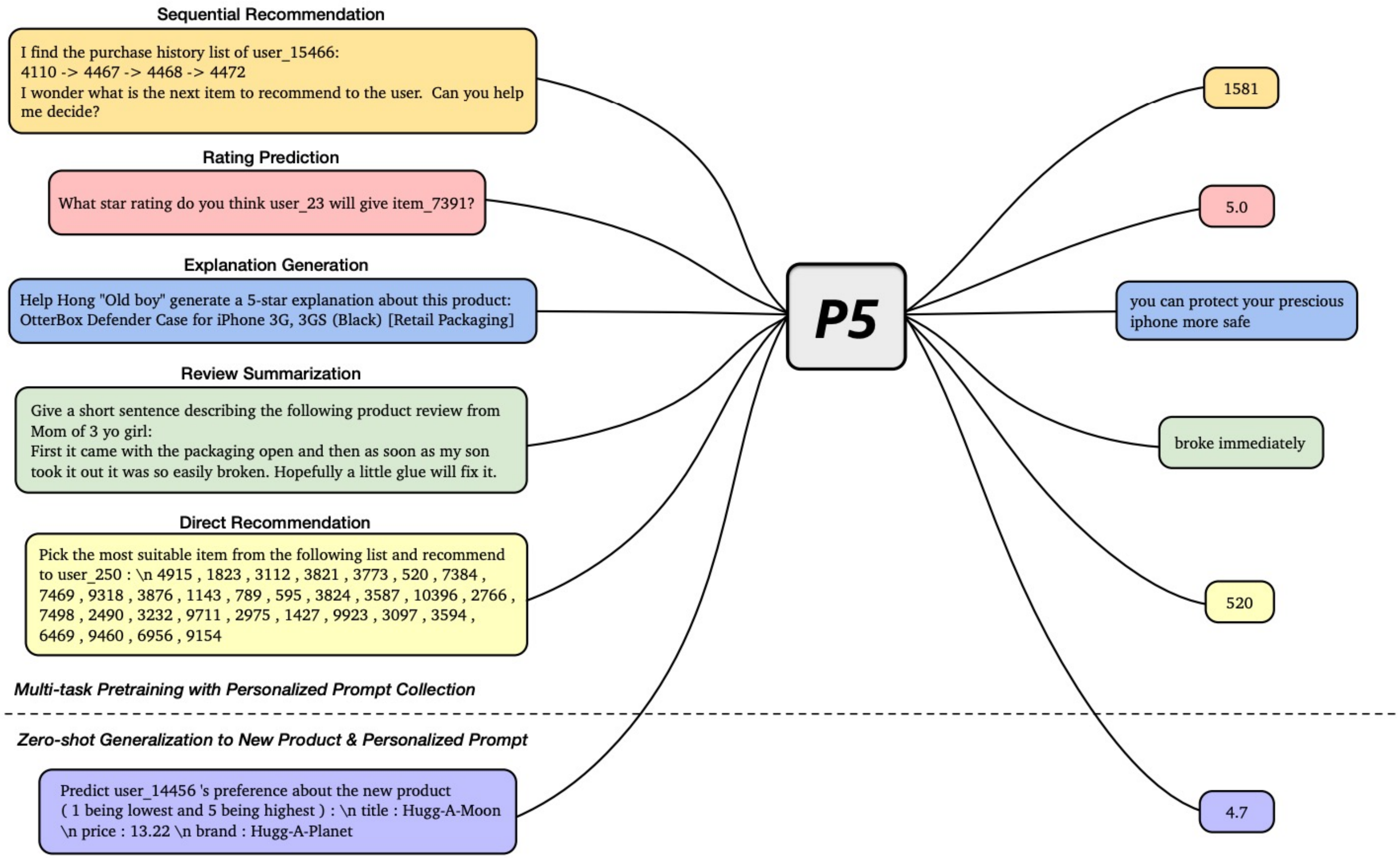




"Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research (2020).

Encoder-Decoder Models for Rec: P5

- Text-to-text paradigm - "Pretrain, Personalized Prompt, and Predict Paradigm" (P5) for recommendation: converting five problems into a text generation problem.



"Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys (2022).

PART 2: Preliminaries of RecSys and LLMs

Website

Website of this tutorial

- **Recommender Systems (RecSys)**
  - Collaborative Filtering (CF)
  - Content-based Recommendation
  - Deep Recommender Systems
- **Large Language Models (LLMs)**
  - Development and Capability
  - LLM Architecture
- **LLM-based RecSys**
  - ID-based LLM RecSys
  - Text-based LLM RecSys

LLM-based RecSys: ID-based & Text-based

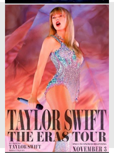
- Users and Items can be represented in various ways

Index or Content ?

○ **User ID**

U8189cf6745fc0d808977bdb0b9f22995

Username: Jack0513

Poster	Movie Name	Numeric ID
	In Broad Daylight	1697292155
	The Marvels	1699436461
	TAYLOR SWIFT | THE ERAS TOUR	1695730583
	The Dark Knight Rises	1699611567
	Oppenheimer	1687513232

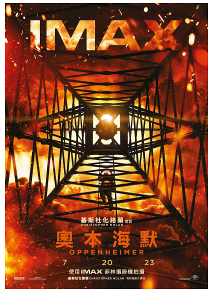
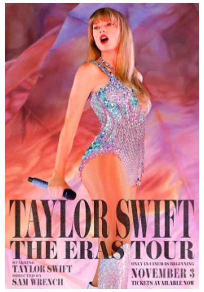
User & Item Representation in LLMs

ID-based LLM RecSys

Text-based LLM RecSys

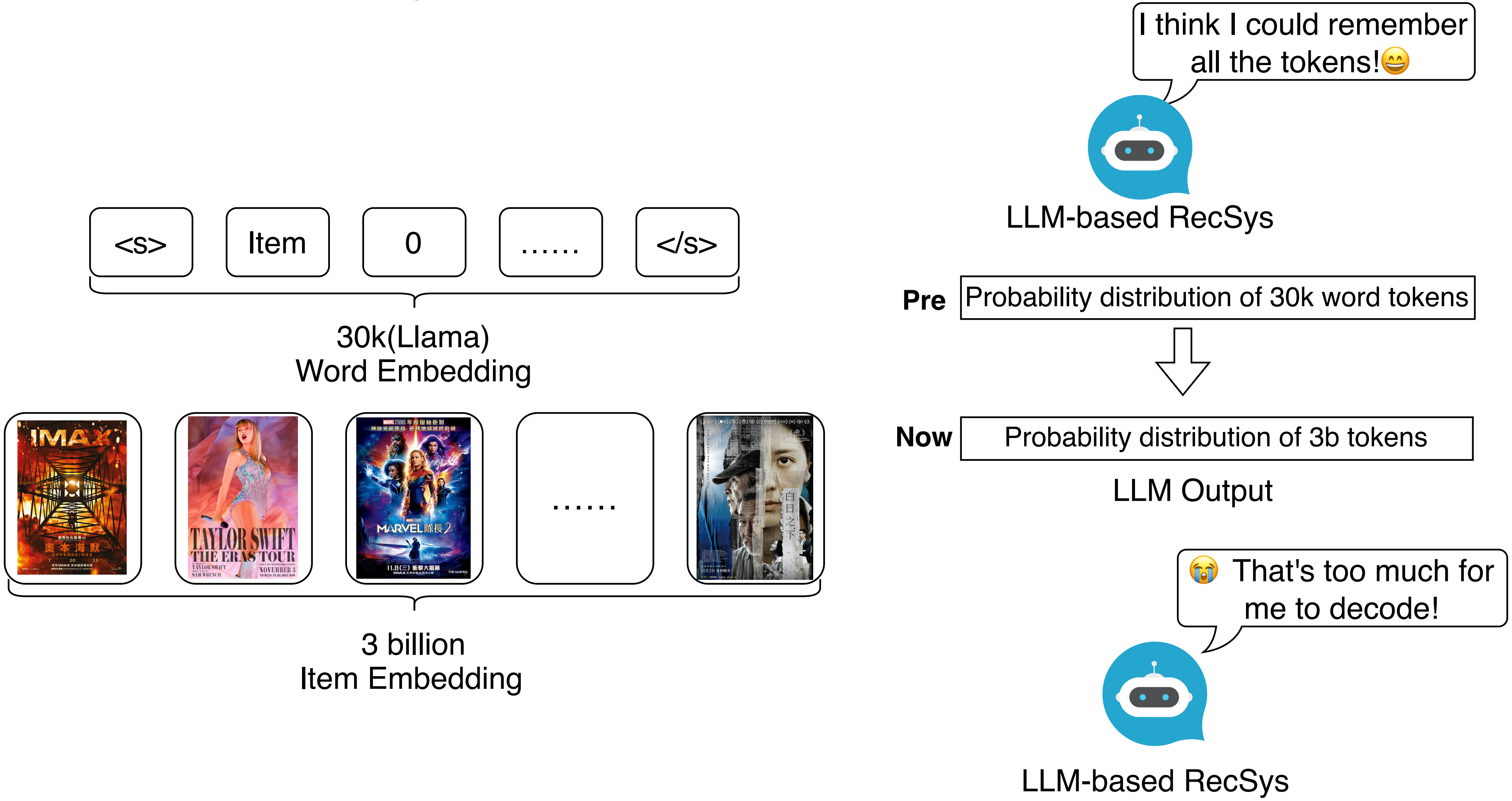
ID-based LLM RecSys

- ❑ Various ways of assigning IDs

	Randomly	Based on Popularity	Based on Time
	AXGGWD027	01	1687513232
	XJSGDG0881	02	1695730583
	BXGW2UD803	03	1699436461

ID-based LLM RecSys

- IDs are originally for unique identification
- However, the embedding of LLMs cannot hold millions of items and users



ID-based LLM RecSys

- ❑ Normally, we can represent users and items with a span of tokens.
- ❑ The format is like "[Prefix]_[ID]". Examples:
  - ❖ User_0123 : ["User", "_", "0", "1", "2", "3"]
  - ❖ Item_5471 : ["Item", "_", "5", "4", "7", "1"]

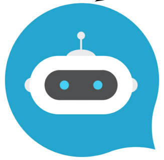
3 billion

⇩

Item_01	Item_02	Item_03	……	Item_30m

Token Spans

😊 No extra decoding cost! I can handle that!

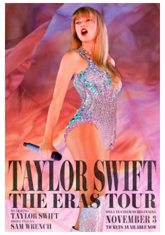


LLM-based RecSys

- ❖ However, for Item_1003, it could be ["Item", "_", "100", "3"], which might be confusing for LLMs!

ID-based LLM RecSys

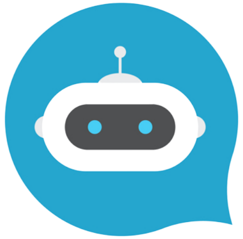
- Indexing methods might affect the performance of RecSys

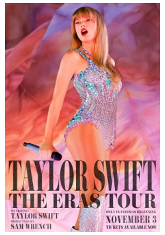
01


02


03

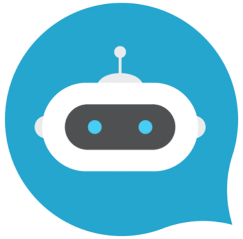
Is Item "04" still a movie?



01


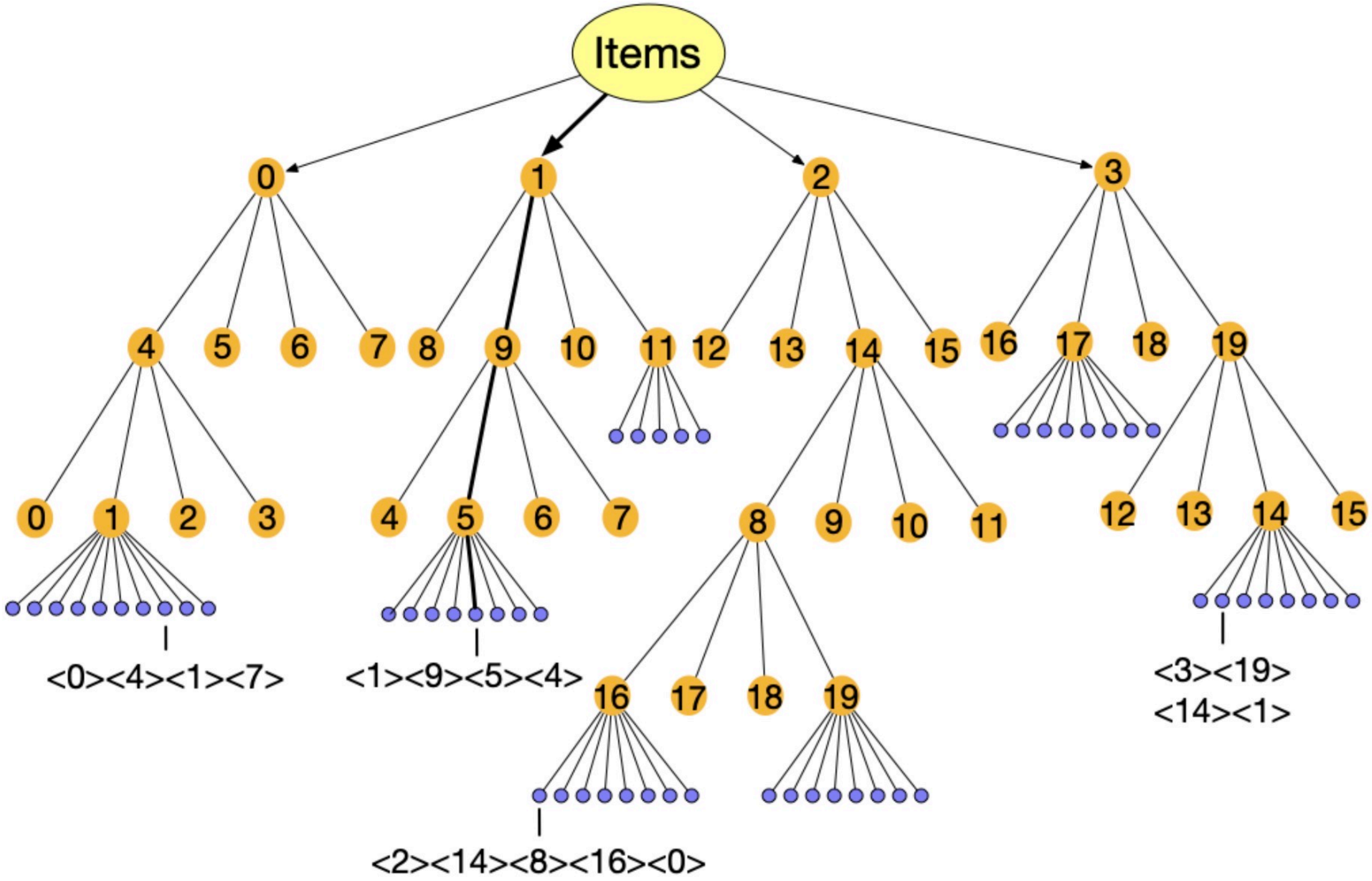
02

Do Item "01" and Item "02" share similar characteristics?



ID-based LLM RecSys

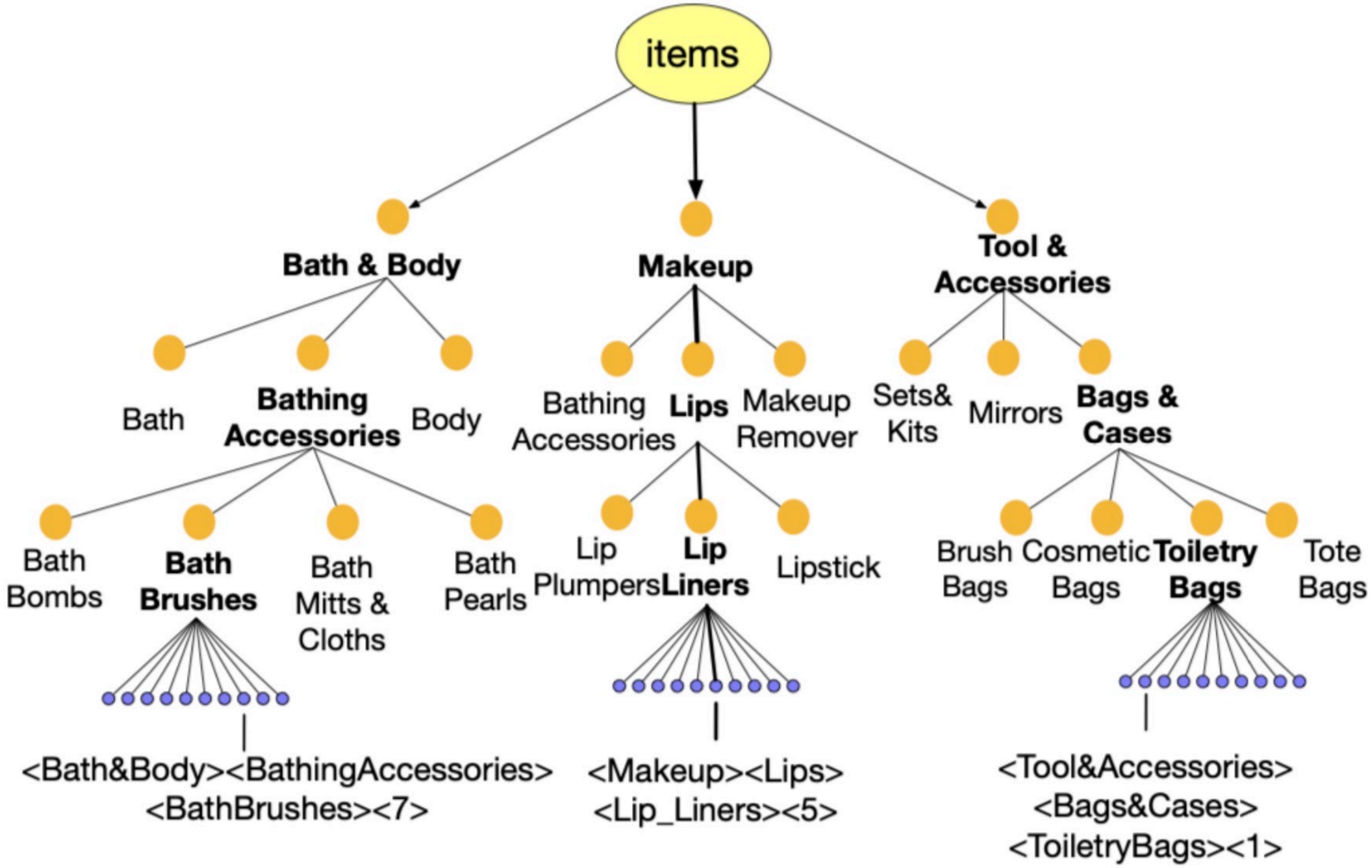
- Introducing more Information to enhance the ID representation
  - Collaborative Indexing



"How to Index Item IDs for Recommendation Foundation Models." arXiv preprint (2023).

ID-based LLM RecSys

- Introducing more Information to enhance the ID representation
  - Semantic Indexing



"How to Index Item IDs for Recommendation Foundation Models." arXiv preprint (2023).

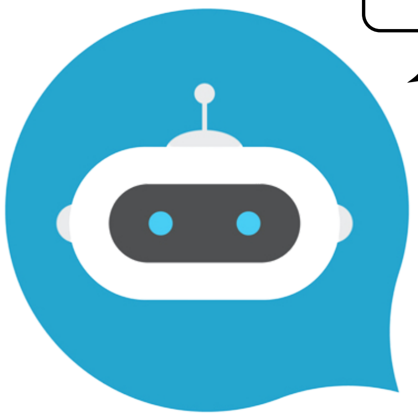
ID-based LLM RecSys

- Modeling user interaction history with Markov chain

User_4782 has bought
3472, 7653, 0192, 4271.
What will he buy next?

Question

User_4782 will buy 7251.



RecSys

ID-based LLM RecSys

- Modelling user interaction history with Markov chain

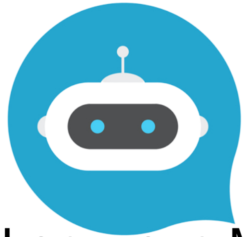
User_0513 bought 3472, 1784, 2563, 8892, 4271, 0988

User_4781 bought 7562, 8892, 4057, 0192, 7251

……

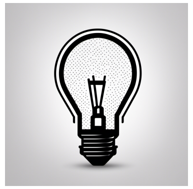
User_2788 bought 4271, 7251, 9082, 0192, 5672

Pre-training & Fine-tuning



Large Language Models

$$v_{i+1} = \arg\max_{v} P(v_{i+1}|v_1, v_2, \dots, v_i)$$



User_4782 will buy 7251.

Modelling the probability of the next item

ID-based LLM RecSys

- The N-gram probability in NLP
  - Unigram

$$P(\text{“3472”}) = \frac{1}{16} \qquad P(\text{“1784”}) = \frac{1}{16}$$

$$P(\text{“2563”}) = \frac{1}{16} \qquad \boxed{P(\text{“8892”}) = \frac{2}{16}}$$

$$\boxed{P(\text{“4271”}) = \frac{2}{16}} \qquad P(\text{“0988”}) = \frac{1}{16}$$

$$P(\text{“7562”}) = \frac{1}{16} \qquad P(\text{“4057”}) = \frac{1}{16}$$

$$\boxed{P(\text{“0192”}) = \frac{2}{16}} \qquad \boxed{P(\text{“7251”}) = \frac{2}{16}}$$

$$P(\text{“9082”}) = \frac{1}{16} \qquad P(\text{“5672”}) = \frac{1}{16}$$

User_0513 bought 3472, 1784, 2563, 8892, 4271, 0988

User_4781 bought 7562, 8892, 4057, 0192, 7251

……

User_2788 bought 4271, 7251, 9082, 0192, 5672

User_4782 has bought 3472, 7653, 0192, 4271. What will he buy next?

Question

ID-based LLM RecSys

- The N-gram probability in NLP
  - Bigram

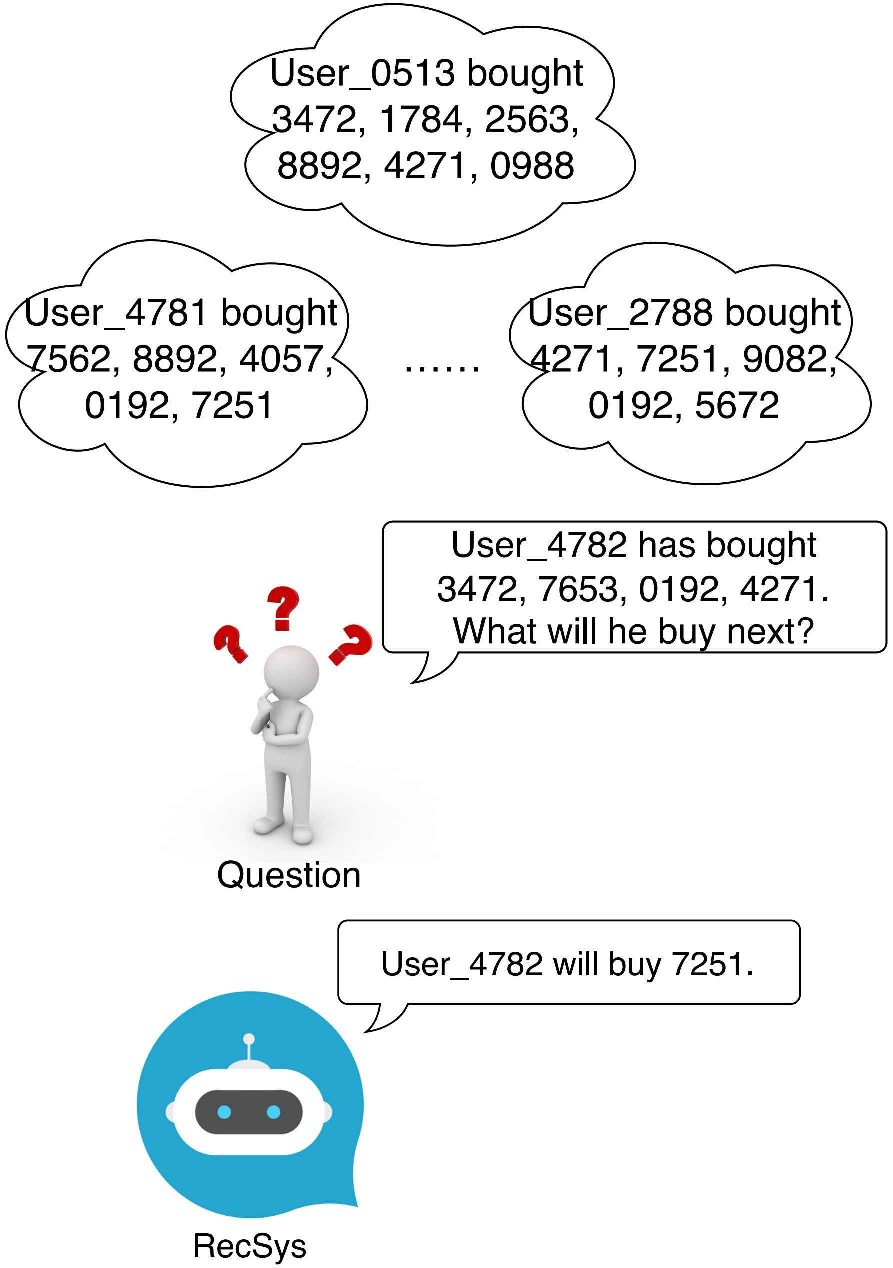
$$P(\text{“0988”} \mid \text{“4271”}) = \frac{1}{2}$$

$$P(\text{“7251”} \mid \text{“4271”}) = \frac{1}{2}$$

  - Which one to choose?

$$P(\text{“0988”}) = \frac{1}{16}$$

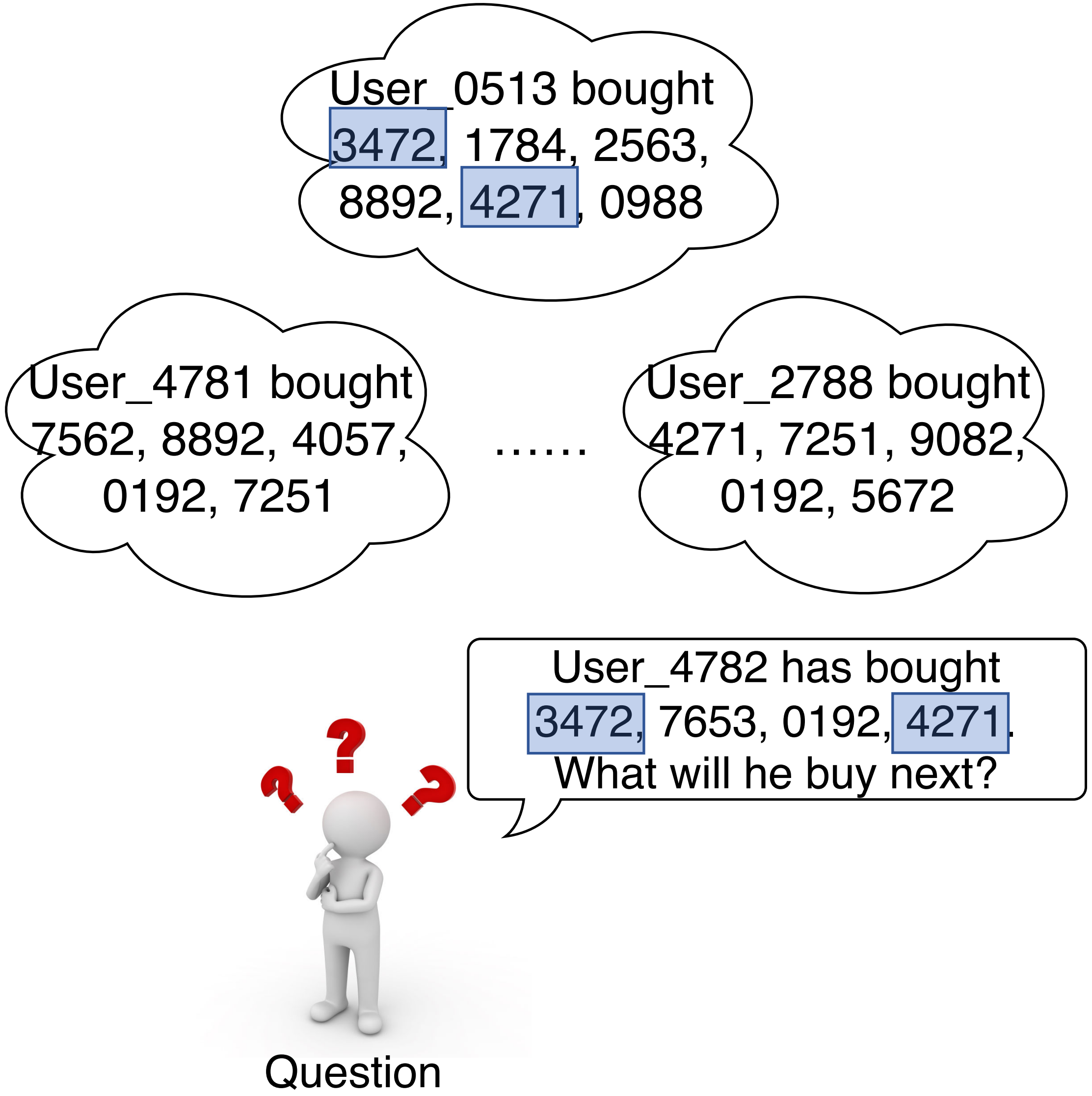
$$P(\text{“7251”}) = \frac{2}{16}$$



ID-based LLM RecSys

- The N-gram probability in NLP
  - The co-occurrence of item IDs
  - User_0513 bought 3472, …, 4271, 0988
  - User_4782 bought 3472, …, 4271, ?

  - Is "0988" a better answer than "7251"?



ID-based LLM RecSys

- Contextual representations of words in LLMs
  - User_0513 bought 3472, 1784, 2563, 8892, 4271, 0988
  - User_4782 bought 3472, 7653, 0192, 4271, ?
  - The item representations can vary for different contexts

User & Item Representation in LLMs

ID-based LLM RecSys

Text-based LLM RecSys

Text-based LLM RecSys

GPT4Rec

- Item **title** contains rich semantic information
- It's a natural way to use **text** to describe items

Previously, the customer has bought

Ben Nye Banana Luxury Face Powder 3.0 oz Makeup Kim Kardashian NEW!!!. Rosallini Women Stainless Steel Extension Eyelash Applicator Tool Fish Tail Clip. Beauty Flawless Makeup Blender Sponge Puff (size 1). Fruit Of The Earth 100% Aloe Vera 24oz Gel Pump.

In the future, the customer wants to buy

Fine-tuned GPT-2

GPT4Rec

Ben Nye Luxury Powders - Banana 1.5oz.

Beautyblender Solid Blendercleanser 1 oz.

Professional 15 Color Concealer Camouflage Makeup Palette.

Pro Beauty Makeup Sponge Blender Flawless Smooth Shaped Water Droplets Puff (Random Color).

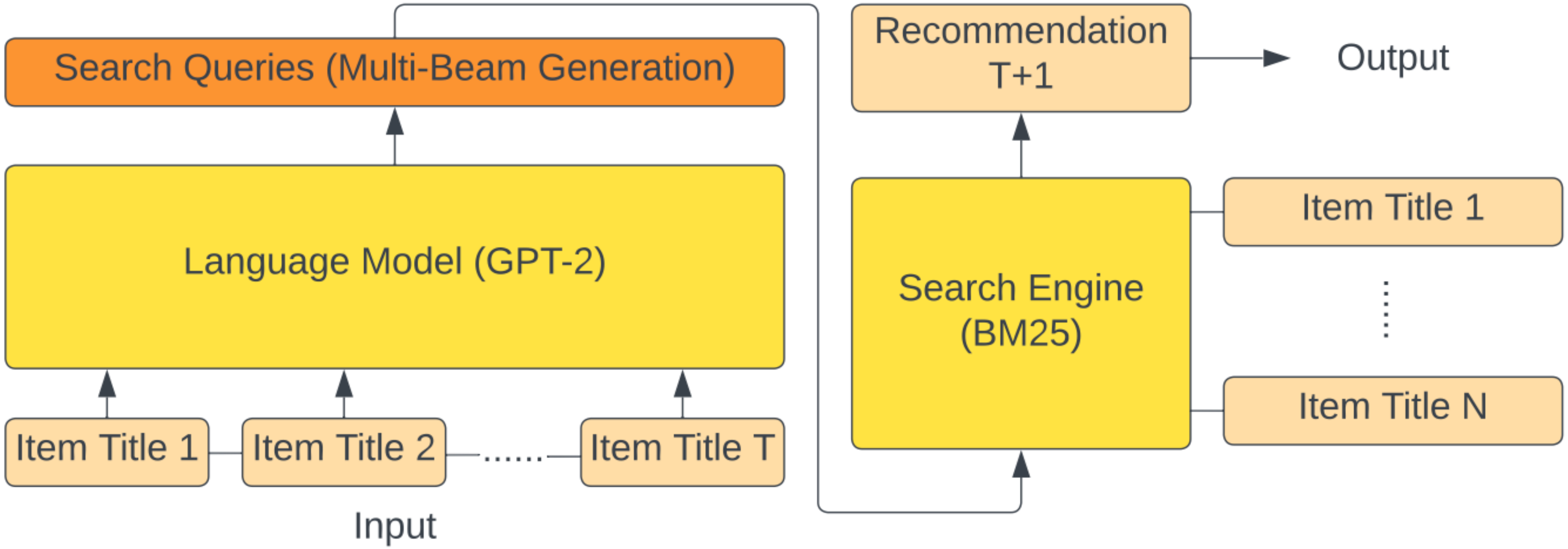
L'Oreal Paris True Match Super Blendable Makeup, Natural Buff, 1.0 Ounces.

"GPT4Rec: A generative framework for personalized recommendation and user interests interpretation." arXiv preprint (2023).

Text-based LLM RecSys

- GPT4Rec

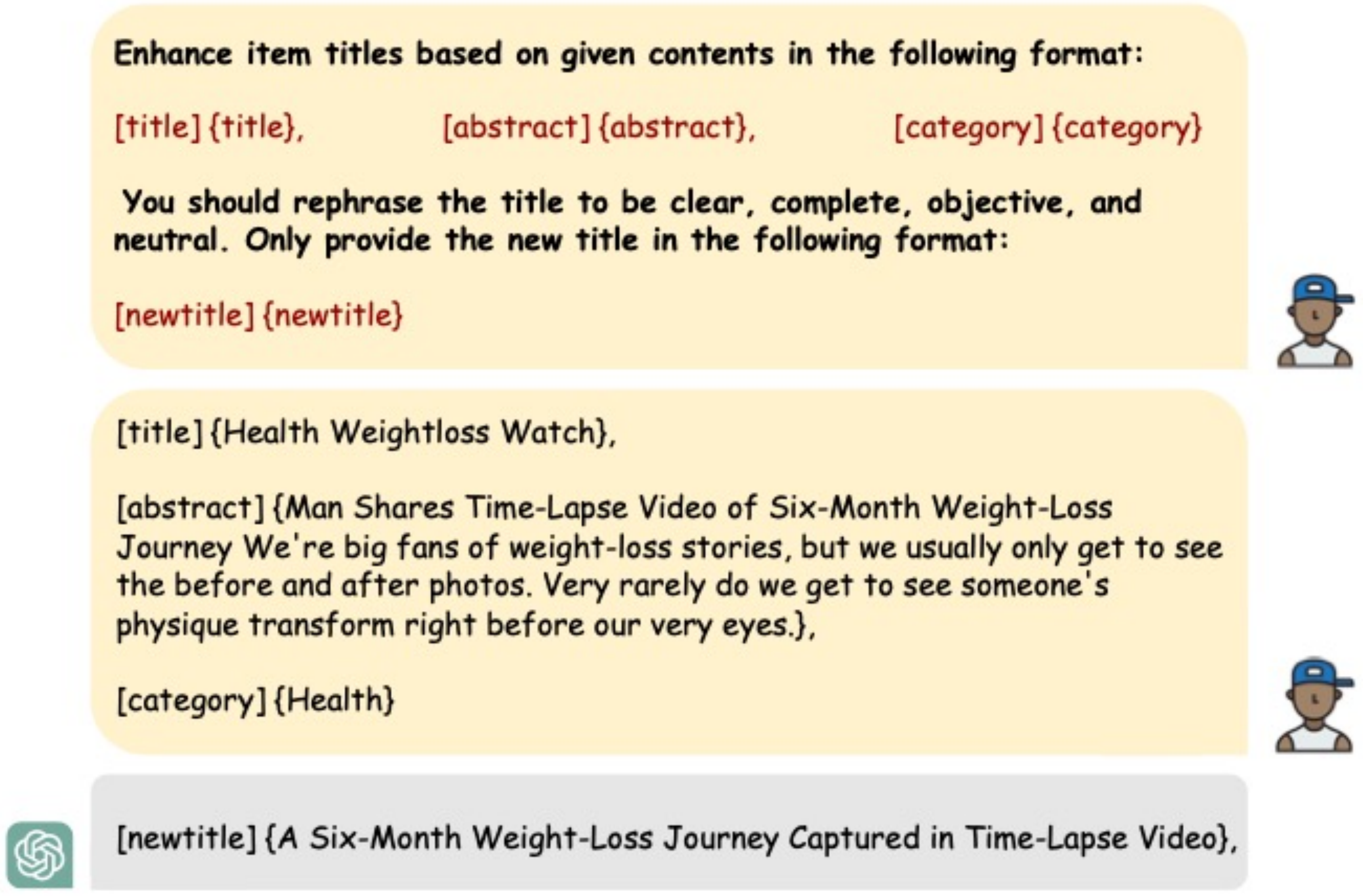
  - In the era of LLMs, **Retrieval-Augmented Generation** (**RAG**) could be a way to improve the capability of LLMs
  - RAG also enhances the explainability of LLM-based RecSys



"GPT4Rec: A generative framework for personalized recommendation and user interests interpretation." arXiv preprint (2023).

Text-based LLM RecSys

- TF-DCon
  - Content-level condensation for recommendation
  - Condense Item title and description to refine item representation

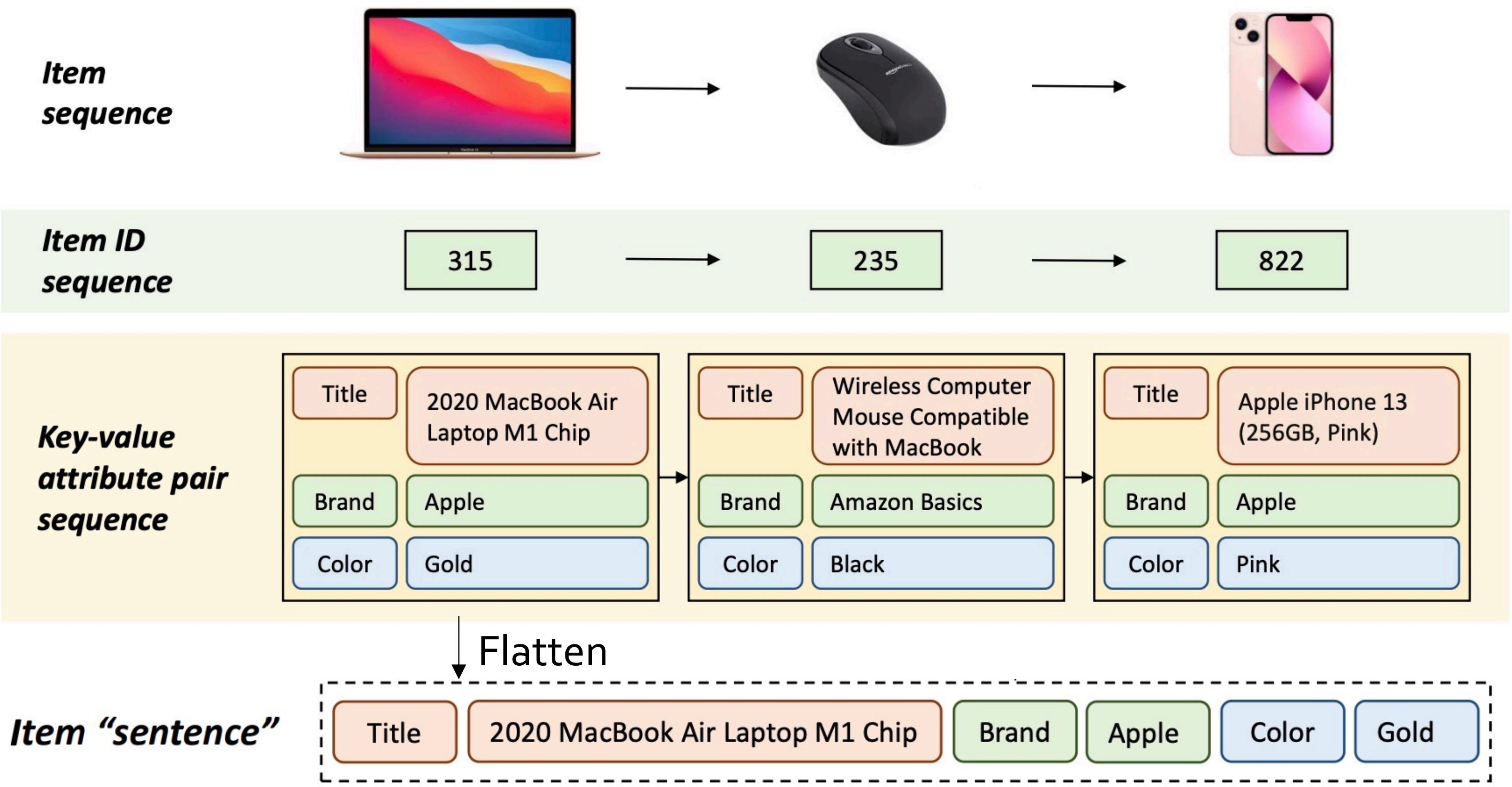


"Leveraging Large Language Models (LLMs) to Empower Training-Free Dataset Condensation for Content-Based Recommendation." arXiv preprint (2023).

Text-based LLM RecSys

❑ Recformer

❖ Use key-value attribute pairs to represent items



" Text Is All You Need: Learning Language Representations for Sequential Recommendation." arXiv preprint (2023).

Tutorial Outline

- **Part 1: Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2: Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3: Pre-training paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)**
- **Part 4: Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5: Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 6: Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website

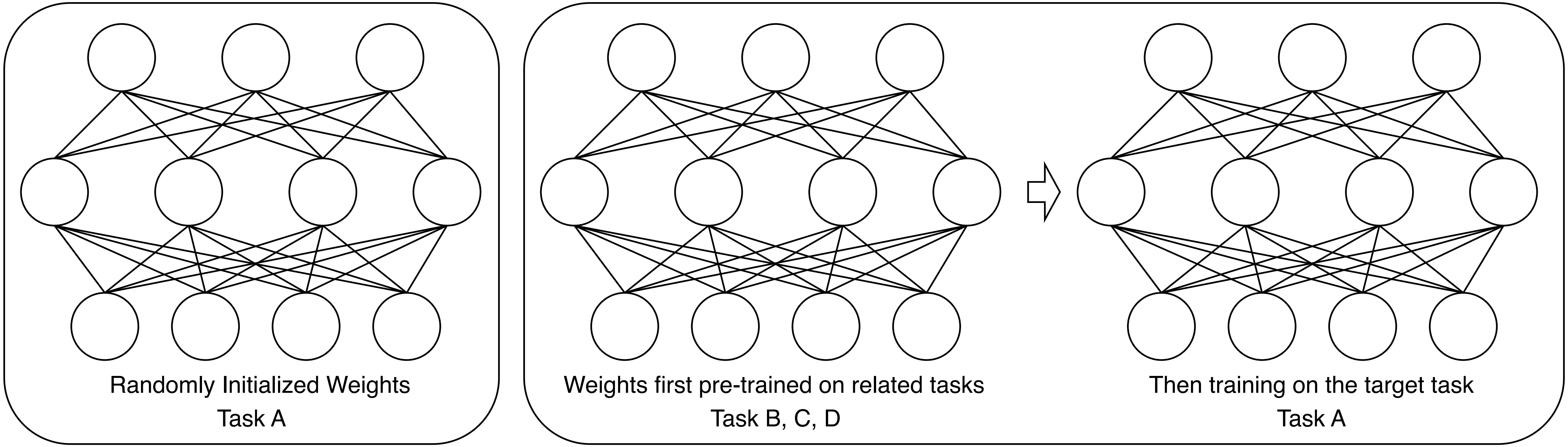
PART 3: RecSys Pre-training

Presenter
Dr. Yujuan DING
HK PolyU

- **Pre-training in NLP**
  - What is pre-training?
  - Why is pre-training needed?
  - NLP pre-training methods
- Pre-training LLM-based RecSys
  - What and why?
  - RecSys pre-training methods

Pre-training in NLP

- What is pre-training?
  - Core Idea: knowledge transfer
  - Technically



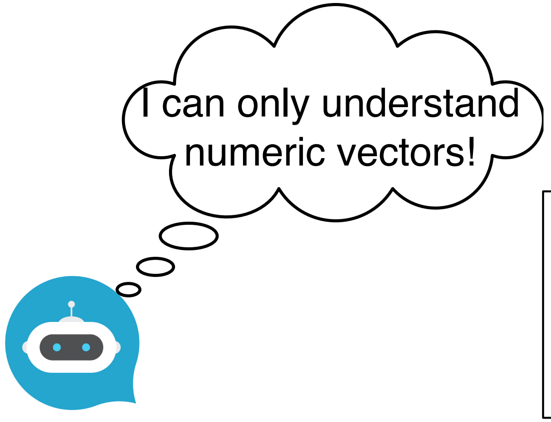
Pre-training in NLP

- Why pre-training?

  - Recall: Tokenization

IJCAI24 is held in Jeju, Korea.
IJCAI24在韩国济州岛举办。

Tokenization discretizes natural language texts



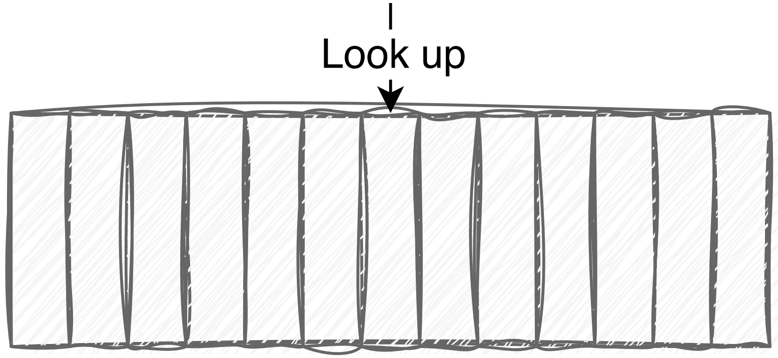
[IJCAI, 24, is, held, in, Jeju, Korea, .]
[IJCAI, 24, 在, 韩国, 济州岛, 举办, 。]

Pre-training in NLP

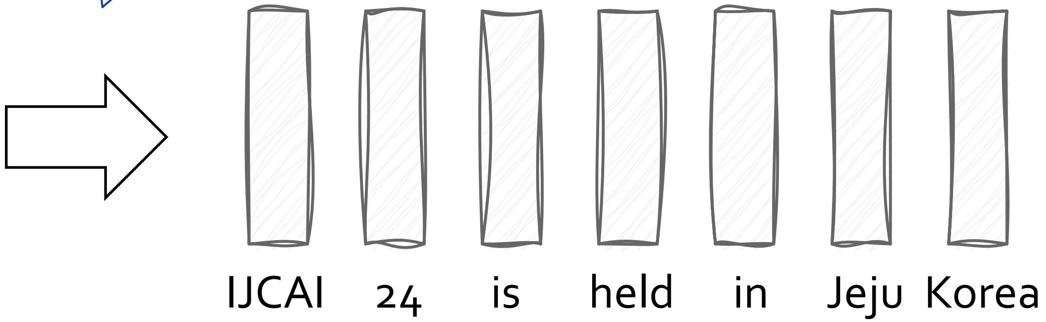
- Why pre-training?
  - Recall: Tokenization

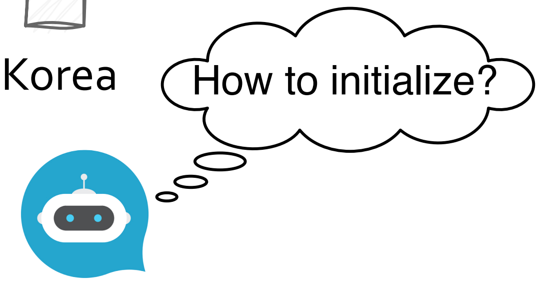
Tokenized natural language texts are mapped to embedding vectors

[IJCAI, 24, is, held, in, Jeju, Korea, .]
[IJCAI, 24, 在, 韩国, 济州岛, 举办, 。]



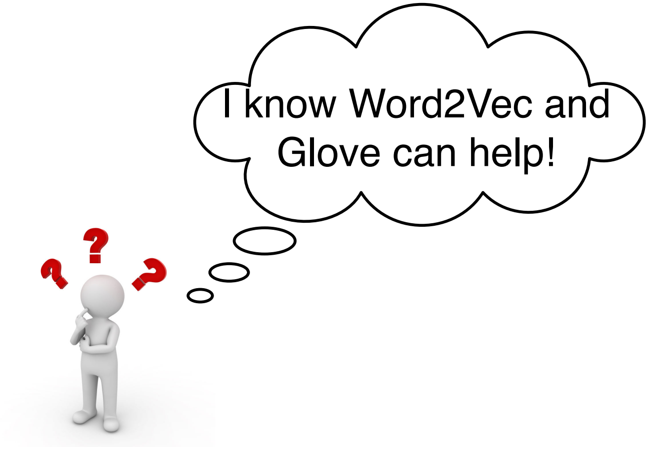
Embedding Matrix

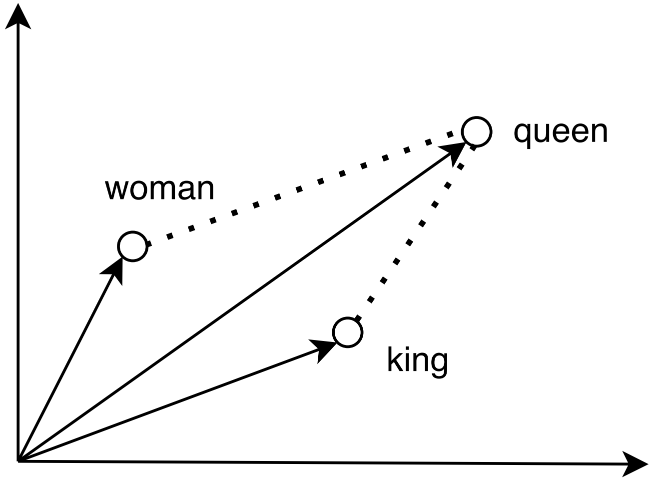




Pre-training in NLP

- Word embeddings?
  - king: [-0.5, -0.9, 1.4, …]
  - queen: [-0.6, -0.8, -0.2, …]
  - woman: [-0.1, -0.1, -1.6, …]





- Static word embeddings (word2vec, Glove) are pre-trained on text corpus from co-occurrence statistics.
  - He is the king of the country

  - She is the queen of the country

Pre-training in NLP

- Problem of static word embedding – Context-Free

He is in charge of the lab.

He and I charge the phone.

Different meanings

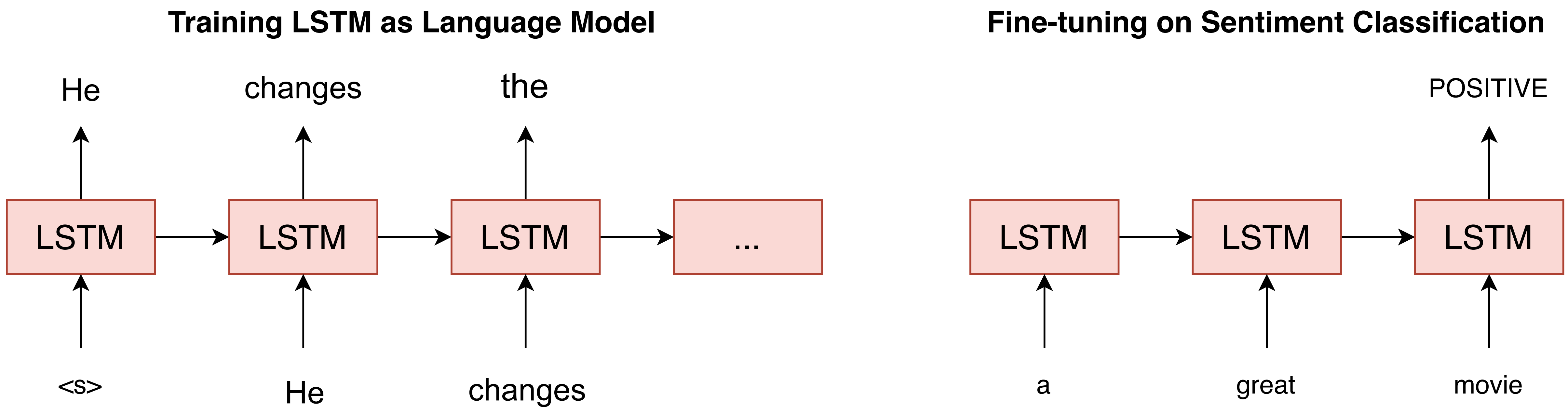
- How to solve it? – Contextual representations

  - He is in charge of the lab
    - charge: [0.2, 0.8, 1.4, ...]

  - He and I charge the phone
    - charge: [-0.3, -0.4, 0.7, ...]

Pre-training in NLP

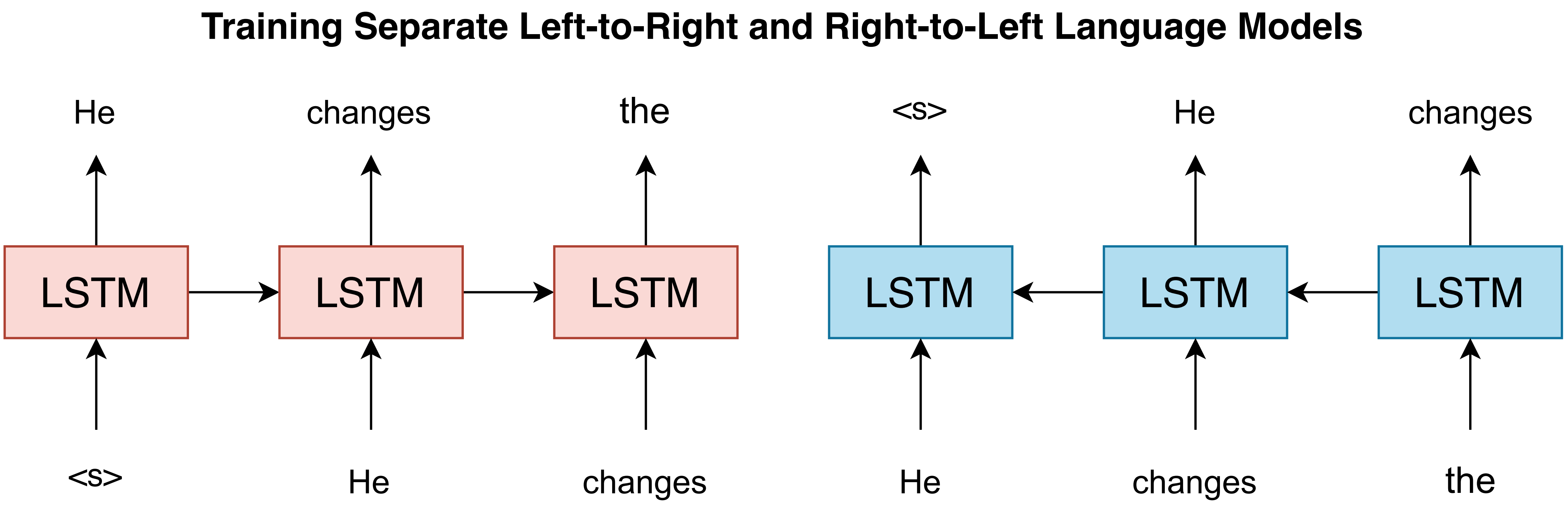
- Semi-Supervised Sequence Learning



"Semi-supervised sequence learning." Advances in neural information processing systems 28 (2015).

Pre-training in NLP

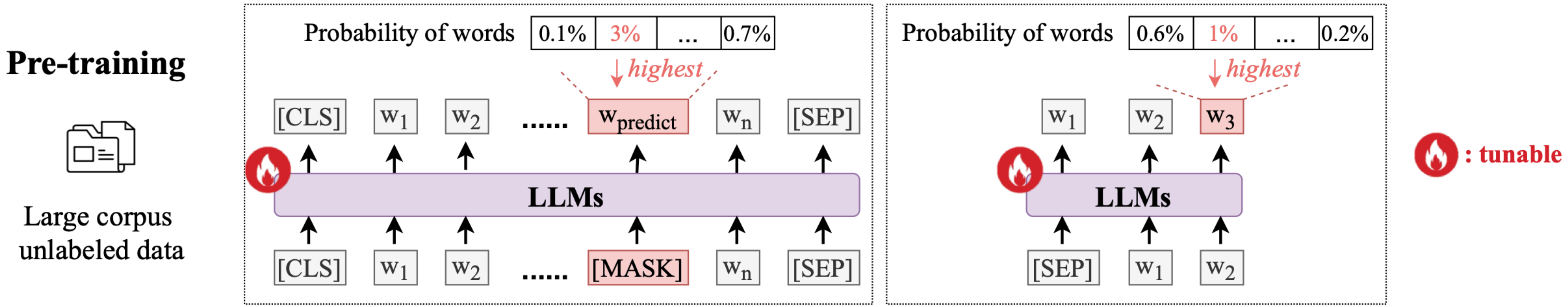
- ELMo: Deep Contextual Word Embeddings



"Deep Contextualized Word Representations." NAACL (2018).

Pre-training in NLP

- Most Favored Pre-training Tasks in NLP
  - Design specific pre-training tasks that could introduce knowledge
    - Masked Language Modelling (For Encoder-Decoder and Encoder-only Structures)
    - Next Token Prediction (For Decoder-only Structures)

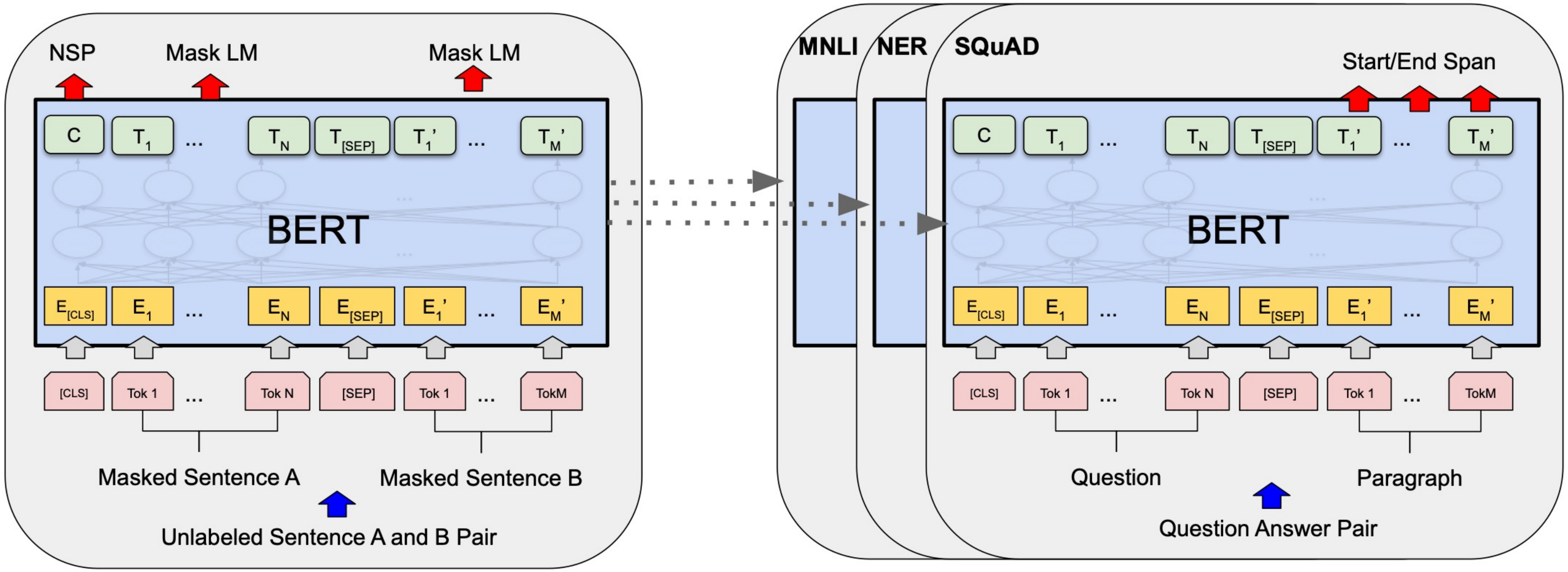


(Masked Language Modeling)

(Next Token Prediction)

Pre-training in NLP

- BERT: Bidirectional Encoder Representations from Transformers



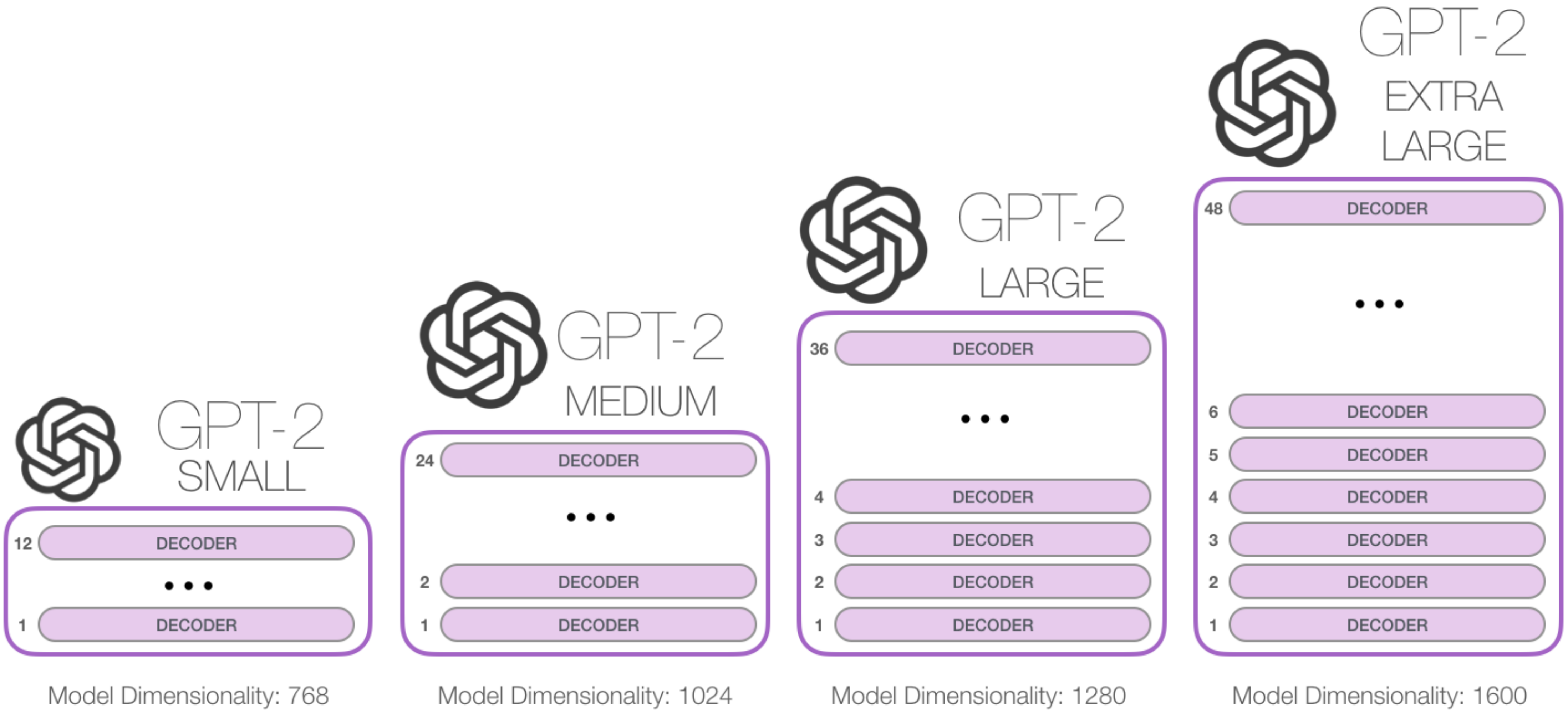
Pre-training

Fine-Tuning

"Bert: Pre-training of deep bidirectional transformers for language understanding." NAACL (2019).

Pre-training in NLP

- GPT



"Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019).

PART 3: RecSys Pre-training

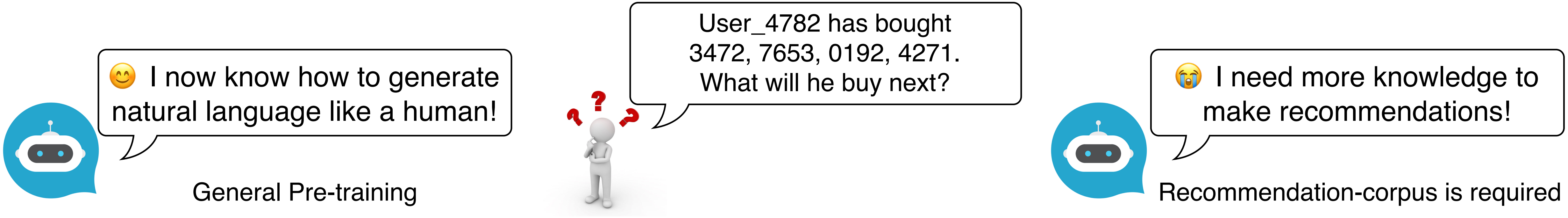
Website of this tutorial

- **Pre-training in NLP**
  - What is pre-training?
  - Why is pre-training needed?
  - NLP pre-training methods
- **Pre-training LLM-based RecSys**
  - What and why?
  - RecSys preotraining methods

Pre-training LLM-based RecSys

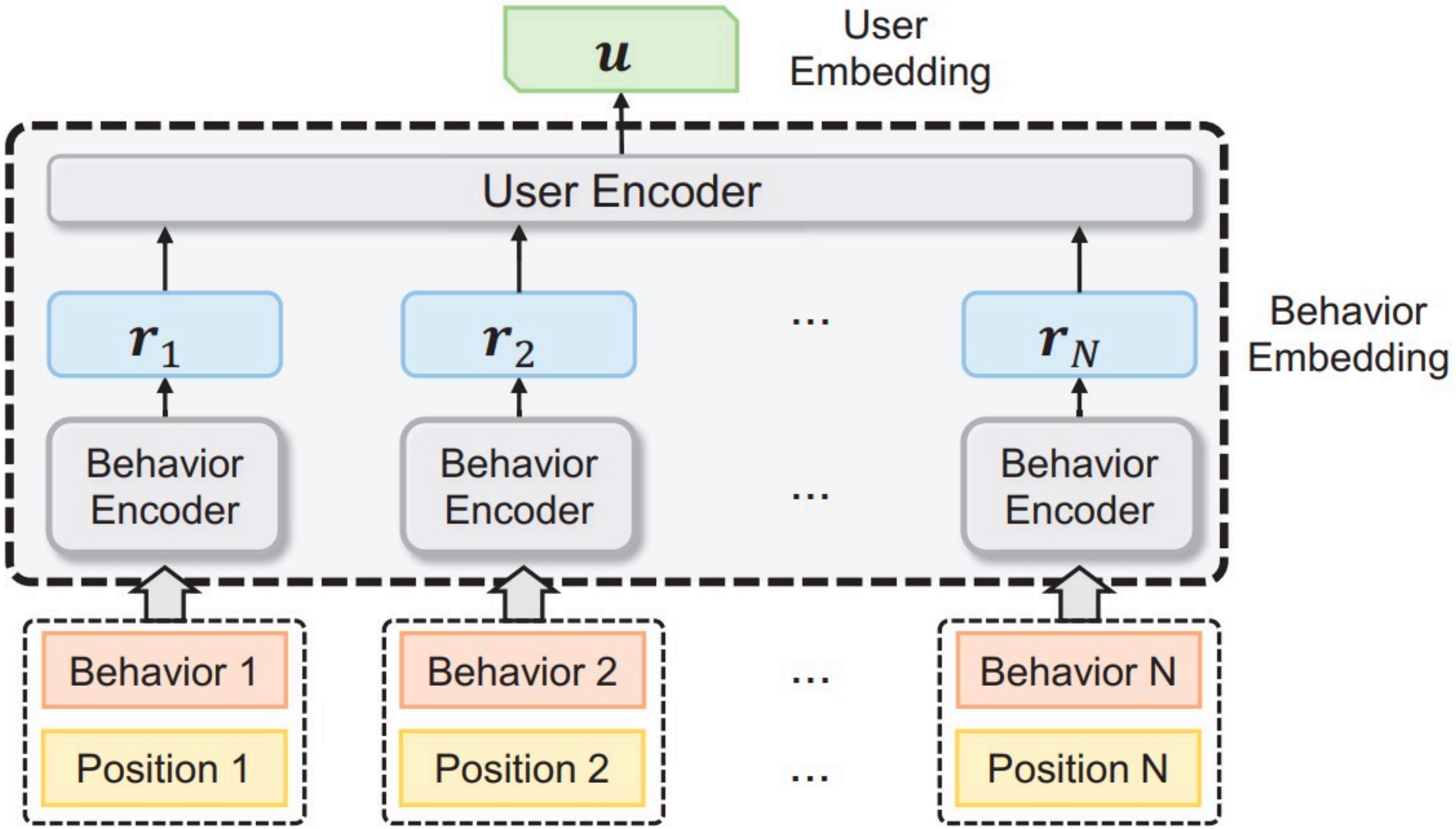
❑ What is Pre-training in LLM-based RecSys and Why is it Necessary?

- ❖ General pre-training vs. domain-specific pre-training
- ❖ Domain knowledge is essential for relieving the knowledge gap



PTUM

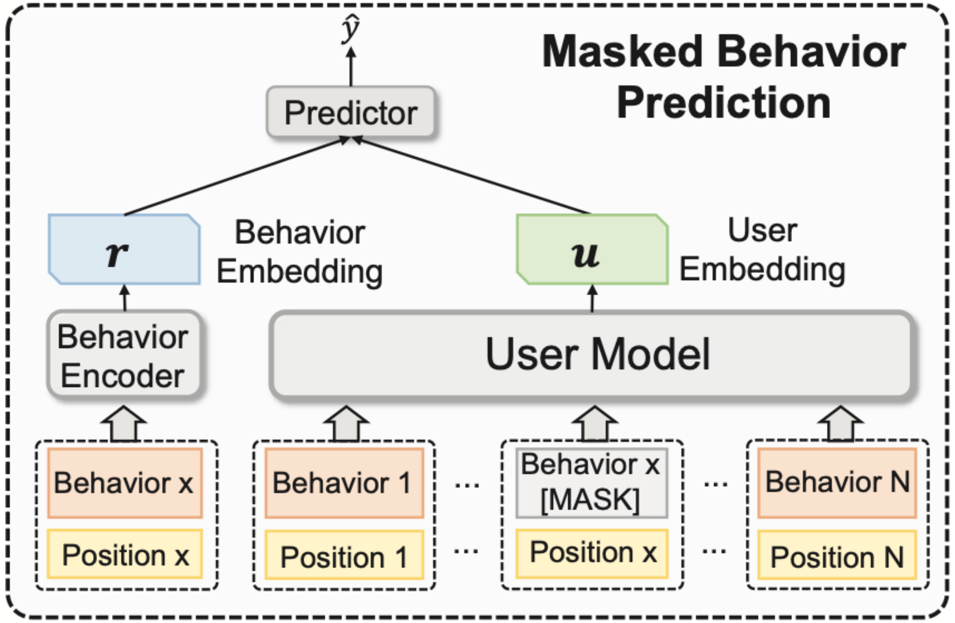
- Pre-training User Model from Unlabeled User Behaviors via Self-supervision



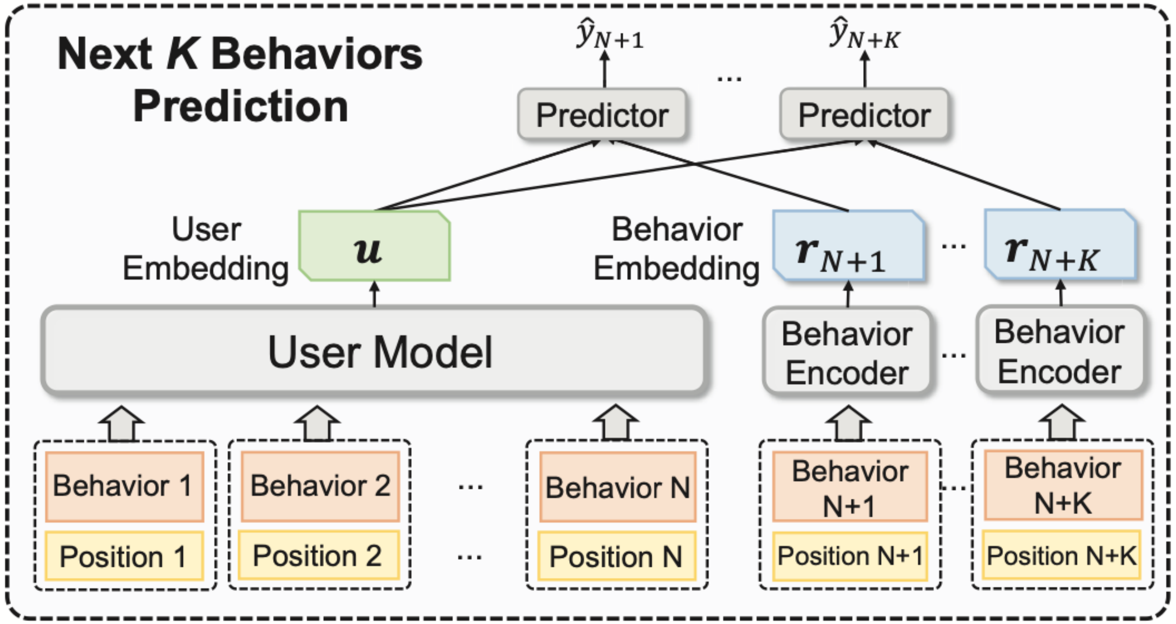
"PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision." EMNLP (2020).

PTUM pre-training tasks

- Masked Behavior Prediction (MBP)
- Next K Behaviors Prediction (NBP)



(a) Masked Behavior Prediction (MBP) task.



(b) Next K Behaviors Prediction (NBP) task.

$$\mathcal{L}_{MBP} = -\sum_{y \in \mathcal{S}_1} \sum_{i=1}^{P+1} y_i \log(\hat{y}_i)$$

$$\mathcal{L}_{NBP} = -\frac{1}{K} \sum_{y \in \mathcal{S}_2} \sum_{k=1}^{K} \sum_{i=1}^{P+1} y_{i,k} \log(\hat{y}_{i,k})$$

$$\mathcal{L} = \mathcal{L}_{MBP} + \lambda \mathcal{L}_{NBP}$$

"PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision." EMNLP (2020).

PTUM application tasks

- ❑ Dataset

Demo			
# users	20,000	avg. # behaviors per user	224.7
# behaviors	4,494,771	avg. # words per webpage title	9.28
CTR			
# users	374,584	avg. # words per webpage title	10.23
# ads	4,159	avg. # words per ad title	11.95
# impressions	400,000	avg. # words per ad description	15.80
# clicked samples	364,281	# non-clicked samples	568,716
# users for pre-training	500,000	# behaviors for pre-training	63,178,293

- ❑ Ads CTR prediction

Methods	20%		50%		100%	
	AUC	AP	AUC	AP	AUC	AP
GRU4Rec	71.45	73.20	71.78	73.85	72.20	74.40
GRU4Rec+PTUM (no finetune)	71.76	73.66	71.95	74.15	72.33	74.77
GRU4Rec+PTUM (finetune)	72.33	74.55	72.42	74.72	72.79	75.40
NativeCTR	71.64	73.47	71.96	74.03	72.35	74.56
NativeCTR+PTUM (no finetune)	71.99	73.95	72.14	74.33	72.50	74.94
NativeCTR+PTUM (finetune)	72.52	74.79	72.59	74.91	72.91	75.57
BERT4Rec	71.82	73.97	72.39	74.89	72.99	75.45
BERT4Rec+PTUM (no finetune)	72.16	74.46	72.58	75.21	73.15	75.83
BERT4Rec+PTUM (finetune)	**72.74**	**75.34**	**73.03**	**75.81**	**73.59**	**76.48**

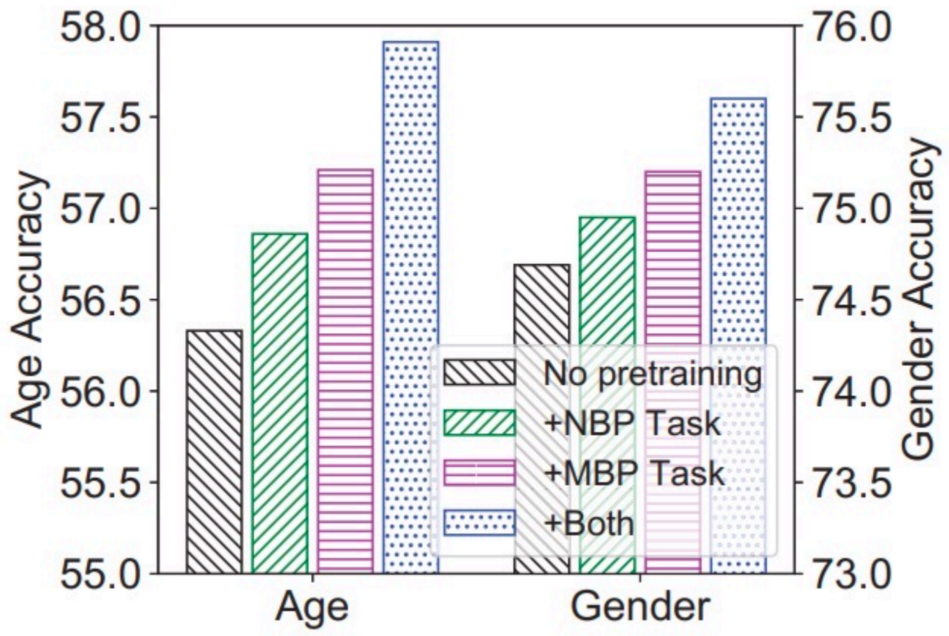
- ➢ Helpful across all datasets and setting scenarios

PTUM effect of pre-training tasks

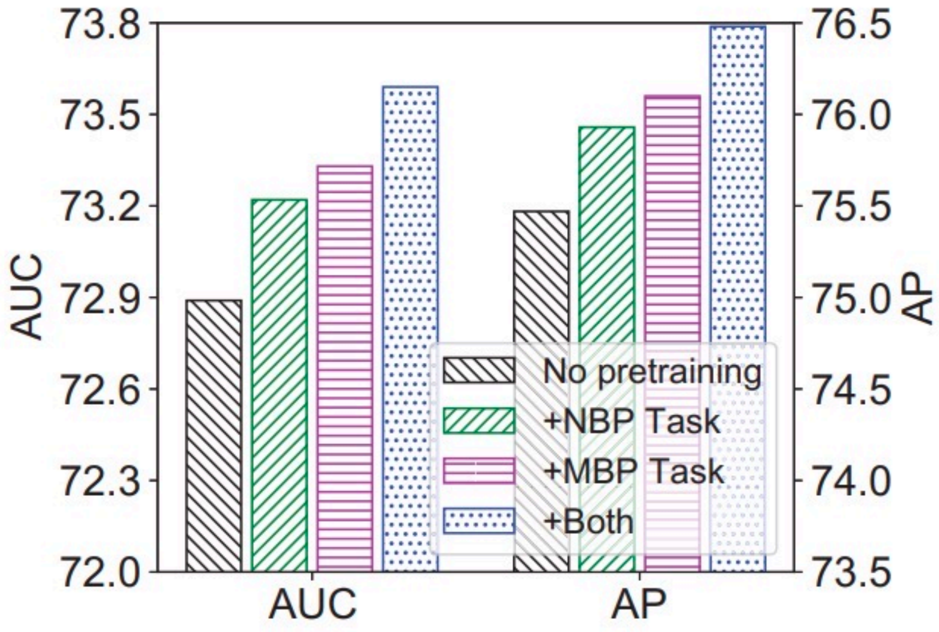
$$\mathcal{L}_{MBP} = -\sum_{y \in \mathcal{S}_1} \sum_{i=1}^{P+1} y_i \log(\hat{y}_i)$$

$$\mathcal{L}_{NBP} = -\frac{1}{K} \sum_{y \in \mathcal{S}_2} \sum_{k=1}^{K} \sum_{i=1}^{P+1} y_{i,k} \log(\hat{y}_{i,k})$$

$$\mathcal{L} = \mathcal{L}_{MBP} + \lambda \mathcal{L}_{NBP}$$



(a) *Demo* Dataset.



(b) *CTR* Dataset.

M6-Rec

- Foundation recommendation model: one model to facilitate diverse domains and a myriad of tasks
- Challenges
  - the potentially unlimited set of downstream domains and tasks
  - the real-world systems' emphasis on computational efficiency
- Backbone: **M6**
  - is a series of visual-linguistic pretrained models
  - supports both Chinese and English
  - is a multi-modal model which aligns well with our plan to incorporate multi-modal features in the future, has achieved widespread success in Alibaba Group's ecosystem when deployed into real-world businesses

"M6: A chinese multimodal pretrainer". arXiv preprint arXiv:2103.00823, 2021.
"M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv preprint (2022).

M6-Rec: task convert

❑ Behavior Modeling as Language Modeling

❖ **Scoring tasks** (estimate the probability of a user clicking or purchasing an item)

[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]

"M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv preprint (2022).

M6-Rec: task convert

❑ Behavior Modeling as Language Modeling

- **Generation tasks** (personalized product design, explainable recommendation, personalized search query generation and conversational recommendation)

[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, . . . [EOS′] [BOS] [EOS]

- **Zero-shot scoring tasks**

[BOS′] A user clicks hiking shoes [EOS′] [BOS] also clicks trekking poles [EOS] [BOS′] A user clicks hiking shoes [EOS′] [BOS] also clicks yoga knee pads [EOS]

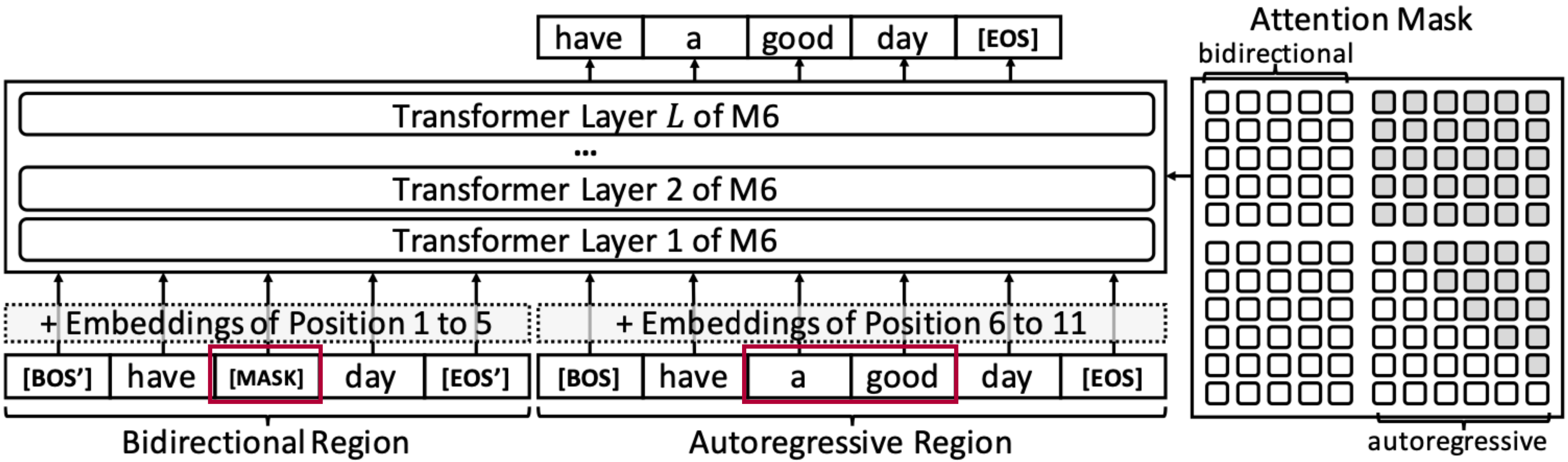
- **Retrieval tasks**

[BOS′] . . . [EOS′] [BOS] The user now purchases a product of category “. . . ” named “. . . ”. Product details: . . . The user likes it because . . . [EOS]

"M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv preprint (2022).

M6-Rec: pre-training task

- Text infilling: masking small spans in a sentence
- Autoregressive language generation: masking the whole sentence



"M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv preprint (2022).

M6-Rec: evaluation task results

- Click-through rate (CTR)

Method	Method Type	Datasets	
		AlipayQuery↑	TaoProduct↑
DIN	ID embeddings	0.7332	0.7611
M6-Rec	Text semantics	**0.7508**	**0.7995**

- Conversational recommendation

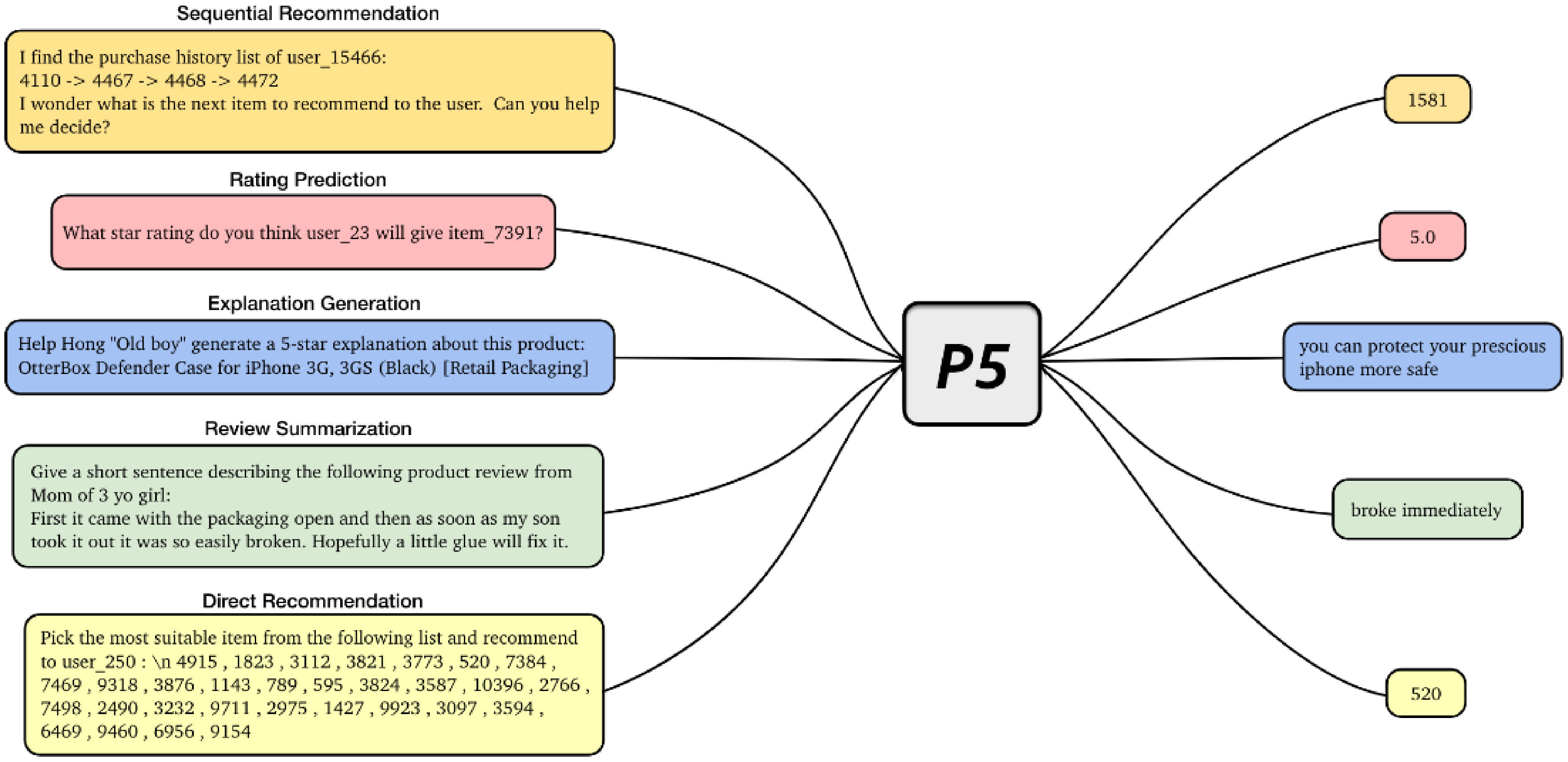
Method	Metrics				
	PPL↓	BLEU-2↑	BLEU-3↑	Dist-3↑	Dist-4↑
Transformer	20.44	0.026	0.014	0.27	0.39
KBRD [3]	17.90	0.060	0.024	0.30	0.45
KGSF [66]	10.73	0.033	0.022	0.40	0.46
M6-Rec	**10.25**	**0.122**	0.021	**0.46**	**0.64**

- kNN retrieval

Method	Method Type	Test Sets	
		All Items↑	Unseen Items↑
YouTubeDNN	ID embeddings	54.4%	*fail*
TwinBERT	Text semantics	69.6%	49.6%
M6-Rec	Text semantics	**74.1%**	**57.0%**

P5

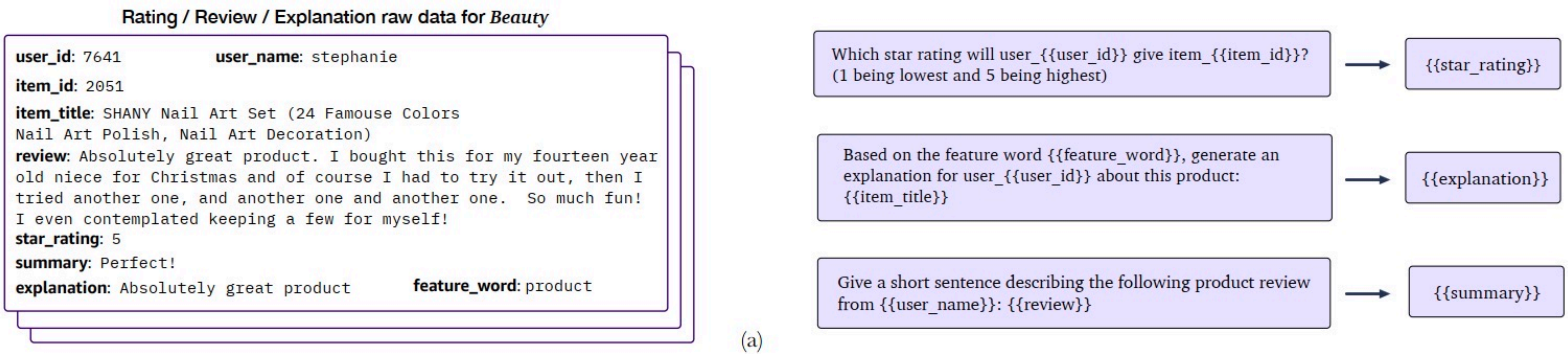
- ❑ Pretrain, Personalized Prompt, and Predict Paradigm
- ❑ Multi-task Pretraining with Personalized Prompt Collection



Multi-task Pretraining with Personalized Prompt Collection

"Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys (2022).

P5: task convert



(a)

P5: task convert

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

Here is the purchase history of user_{{user_id}}:
{{purchase_history}}
What to recommend next for the user?

→ {{next_item}}

(b)

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

Choose the best item from the candidates to recommend for {{user_name}}? \n {{candidate_items}}

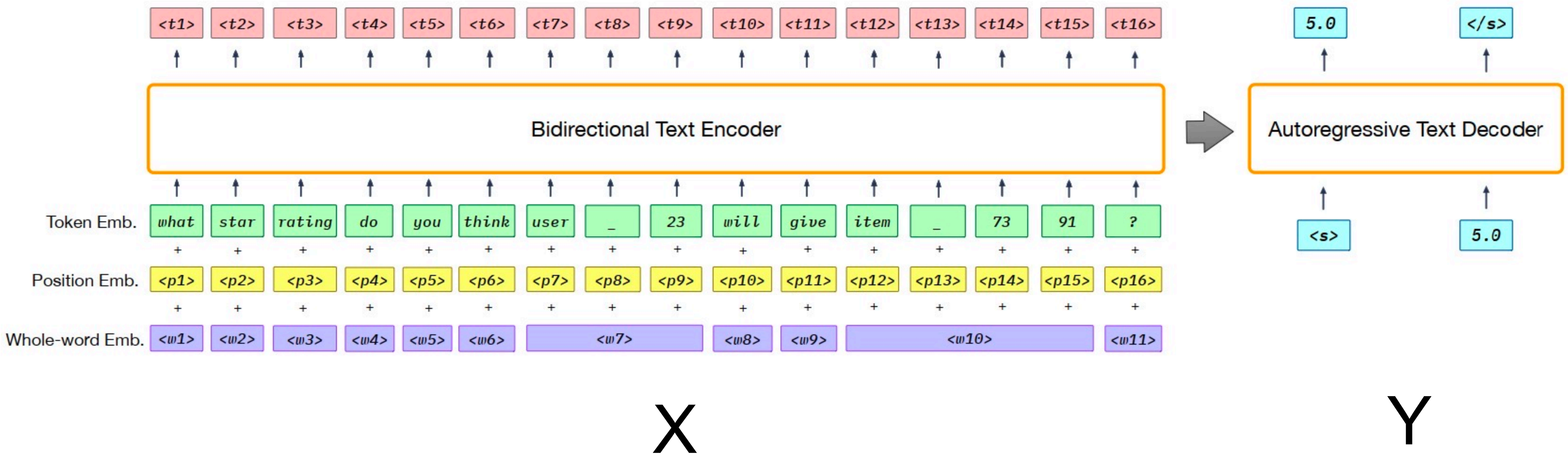
→ {{target_item}}

(c)

P5: pre-training task

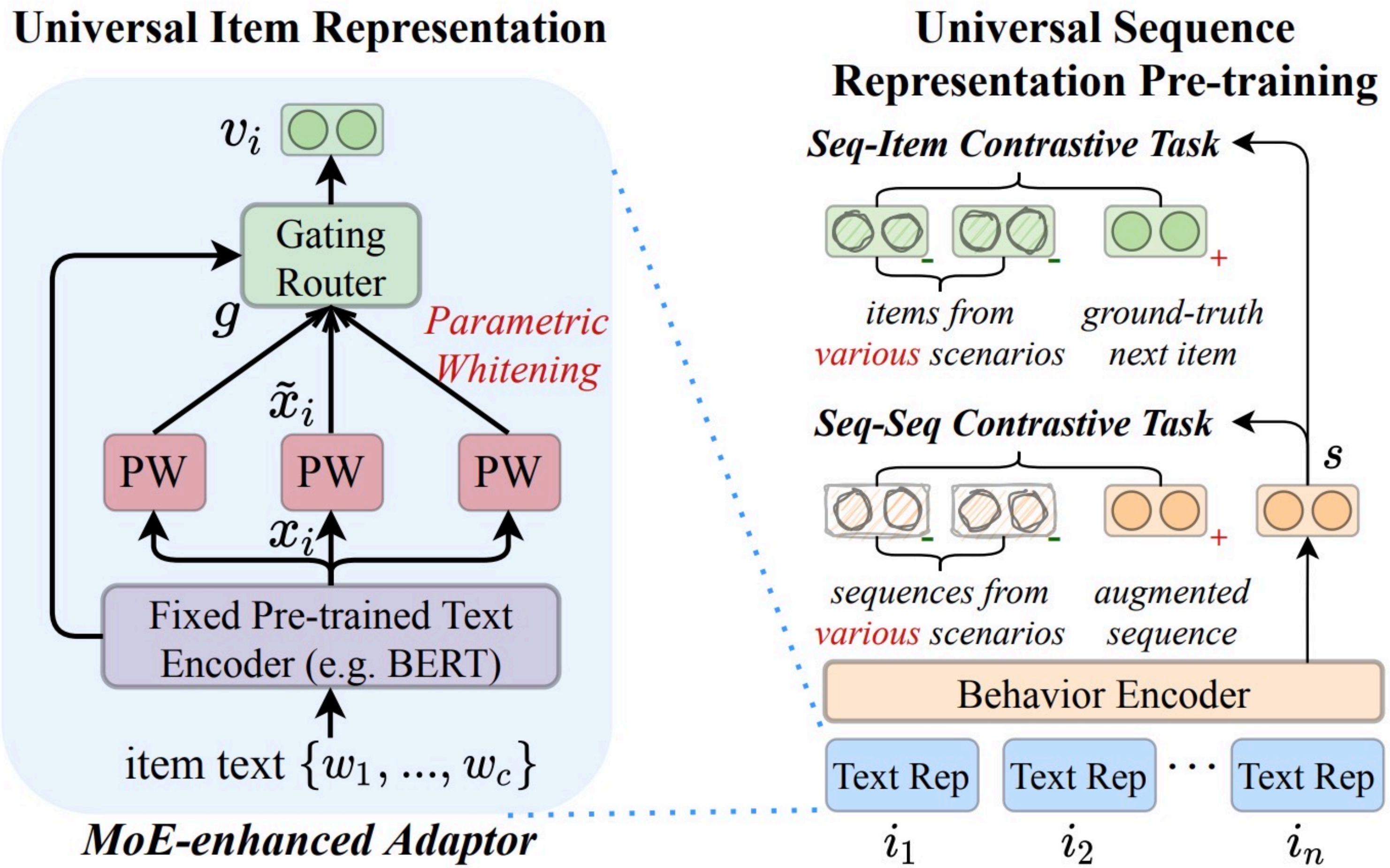
- Label token prediction

$$\mathcal{L}_{\theta}^{\mathrm{P5}} = -\sum_{j=1}^{|\mathbf{y}|} \log P_{\theta}\left(\mathbf{y}_j \mid \mathbf{y}_{<j}, \mathbf{x}\right)$$



UniSRec: universal sequence representation learning

- Utilizing the **associated description text** of items to learn **transferable** representations across different recommendation scenarios.

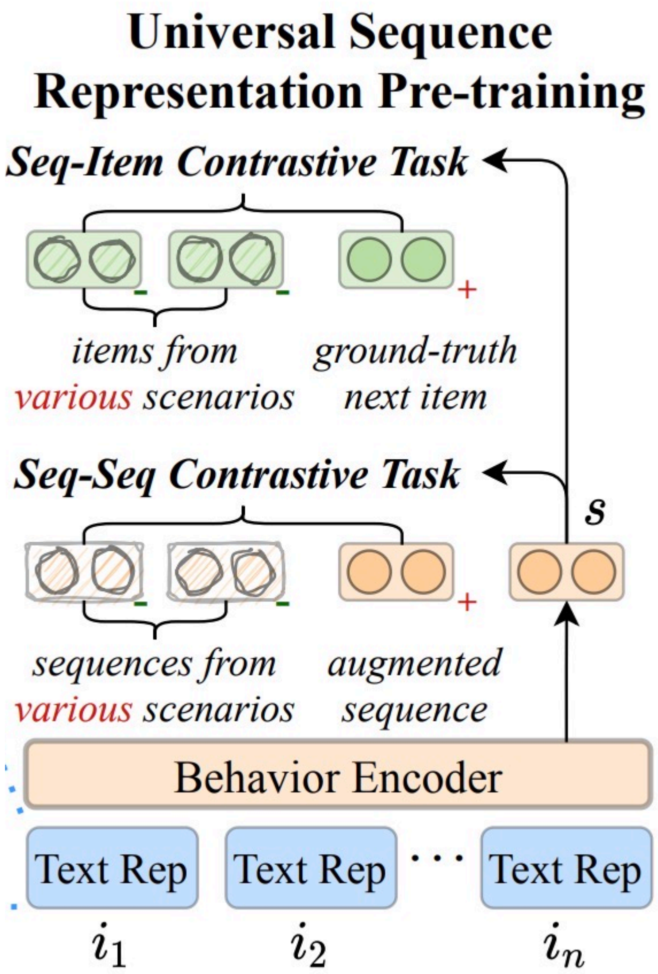


Mixture-of-Expert

Towards Universal Sequence Representaion Learning for Recommendation Systems." KDD. 2022

UniSRec: Pre-training task

- Sequence-item contrastive learning
- Sequence-sequence contrastive learning



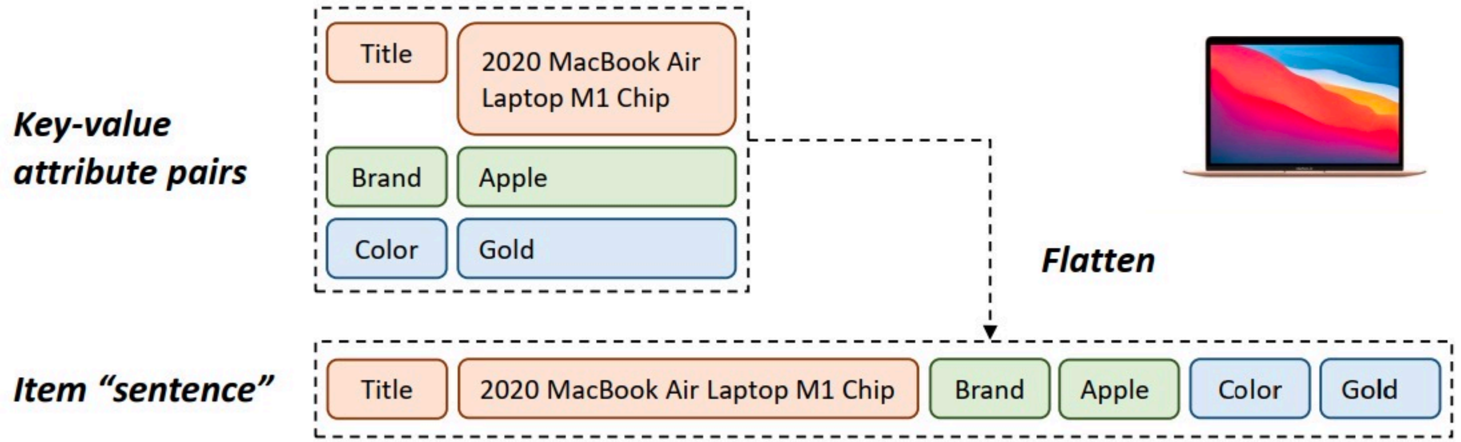
$$\ell_{S-I} = -\sum_{j=1}^{B} \log \frac{\exp\left(\boldsymbol{s}_j \cdot \boldsymbol{v}_j / \tau\right)}{\sum_{j'=1}^{B} \exp\left(\boldsymbol{s}_j \cdot \boldsymbol{v}_{j'} / \tau\right)}$$

$$\ell_{S-S} = -\sum_{j=1}^{B} \log \frac{\exp\left(\boldsymbol{s}_j \cdot \widetilde{\boldsymbol{s}}_j / \tau\right)}{\sum_{j'=1}^{B} \exp\left(\boldsymbol{s}_j \cdot \boldsymbol{s}_{j'} / \tau\right)}$$

$$\mathcal{L}_{\mathrm{PT}} = \ell_{S-I} + \lambda \cdot \ell_{S-S}$$

Recformer

- Item → “sentence” (word sequence): **flattening item key-value attributes** described by text so that an item sequence for a user becomes a sequence of sentences.

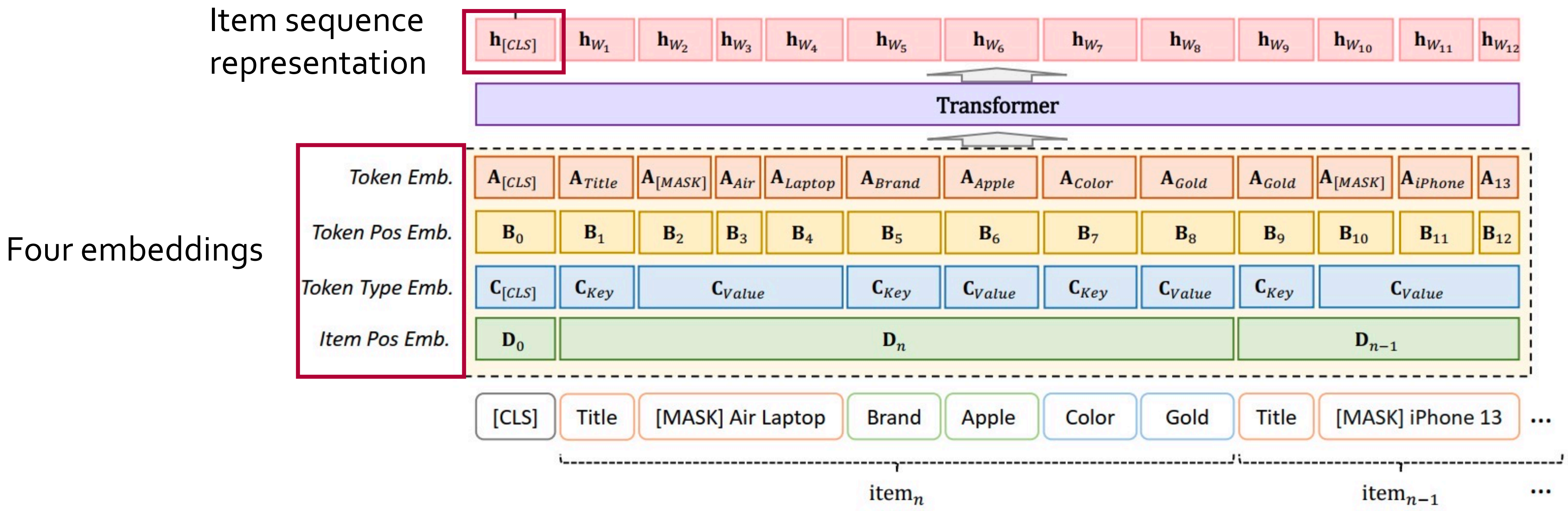


Item “sentence”

"Text is all you need: Learning language representations for sequential recommendation." KDD 2023.

Recformer: model structure

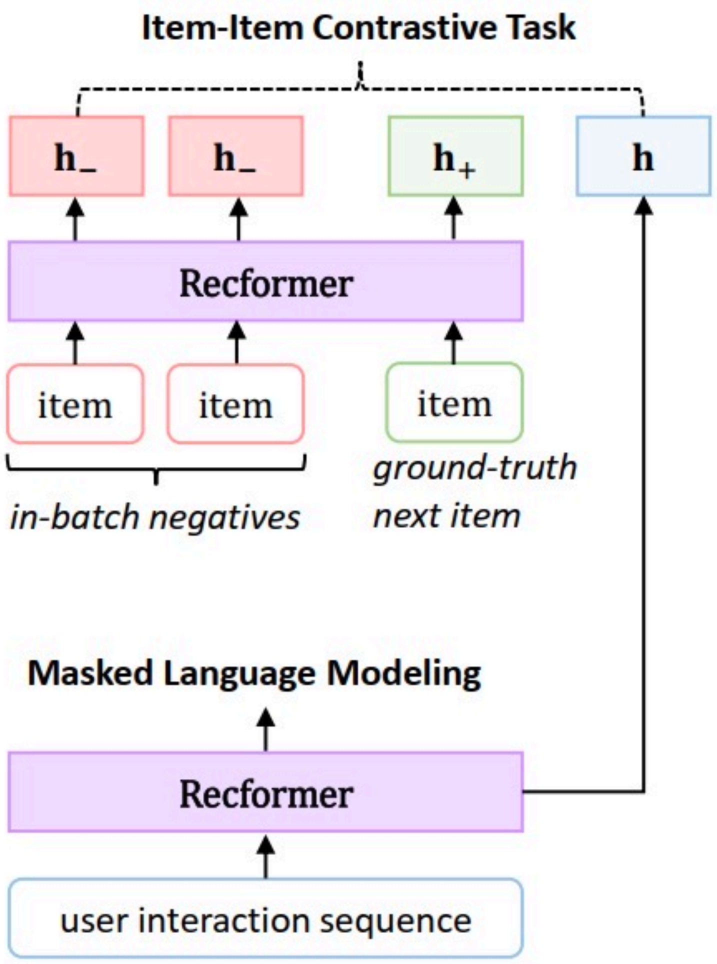
- A similar structure as Longformer:a **multi-layer bidirectional Transformer** with an attention mechanism that scales linearly with sequence length.
- Considering computational efficiency, but also open to other bidirectional Transformer structures such as BERT.



"Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150* (2020).

Recformer: pre-training task

- ❑ Masked Language Modeling (MLM)
- ❑ Item-item contrastive task (IIC)



$$\mathcal{L}_{\mathrm{IIC}} = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_s,\mathbf{h}_i^+)/\tau}}{\sum_{i\in\mathcal{B}} e^{\mathrm{sim}(\mathbf{h}_s,\mathbf{h}_i)/\tau}}$$

$$\mathcal{L}_{\mathrm{MLM}} = -\sum_{i=0}^{|\mathcal{V}|} y_i \log(p_i)$$

$$\mathcal{L}_{\mathrm{PT}} = \mathcal{L}_{\mathrm{IIC}} + \lambda \cdot \mathcal{L}_{\mathrm{MLM}}$$

Recformer: performance

- Different categories of Amazon review datasets

		ID-Only Methods				ID-Text Methods		Text-Only Methods			Improv.
Dataset	**Metric**	**GRU4Rec**	**SASRec**	**BERT4Rec**	**RecGURU**	**FDSA**	**S^3-Rec**	**ZESRec**	**UniSRec**	**Recformer**	
Scientific	NDCG@10	0.0826	0.0797	0.0790	0.0575	0.0716	0.0451	0.0843	0.0862	**0.1027**	19.14%
	Recall@10	0.1055	0.1305	0.1061	0.0781	0.0967	0.0804	0.1260	0.1255	**0.1448**	10.96%
	MRR	0.0702	0.0696	0.0759	0.0566	0.0692	0.0392	0.0745	0.0786	**0.0951**	20.99%
Instruments	NDCG@10	0.0633	0.0634	0.0707	0.0468	0.0731	0.0797	0.0694	0.0785	**0.0830**	4.14%
	Recall@10	0.0969	0.0995	0.0972	0.0617	0.1006	0.1110	0.1078	**0.1119**	0.1052	-
	MRR	0.0707	0.0577	0.0677	0.0460	0.0748	0.0755	0.0633	0.0740	**0.0807**	6.89%
Arts	NDCG@10	0.1075	0.0848	0.0942	0.0525	0.0994	0.1026	0.0970	0.0894	**0.1252**	16.47%
	Recall@10	0.1317	0.1342	0.1236	0.0742	0.1209	0.1399	0.1349	0.1333	**0.1614**	15.37%
	MRR	0.1041	0.0742	0.0899	0.0488	0.0941	0.1057	0.0870	0.0798	**0.1189**	12.49%
Office	NDCG@10	0.0761	0.0832	0.0972	0.0500	0.0922	0.0911	0.0865	0.0919	**0.1141**	17.39%
	Recall@10	0.1053	0.1196	0.1205	0.0647	0.1285	0.1186	0.1199	0.1262	**0.1403**	9.18%
	MRR	0.0731	0.0751	0.0932	0.0483	0.0972	0.0957	0.0797	0.0848	**0.1089**	12.04%
Games	NDCG@10	0.0586	0.0547	0.0628	0.0386	0.0600	0.0532	0.0530	0.0580	**0.0684**	8.92%
	Recall@10	0.0988	0.0953	0.1029	0.0479	0.0931	0.0879	0.0844	0.0923	**0.1039**	0.97%
	MRR	0.0539	0.0505	0.0585	0.0396	0.0546	0.0500	0.0505	0.0552	**0.0650**	11.11%
Pet	NDCG@10	0.0648	0.0569	0.0602	0.0366	0.0673	0.0742	0.0754	0.0702	**0.0972**	28.91%
	Recall@10	0.0781	0.0881	0.0765	0.0415	0.0949	0.1039	0.1018	0.0933	**0.1162**	11.84%
	MRR	0.0632	0.0507	0.0585	0.0371	0.0650	0.0710	0.0706	0.0650	**0.0940**	32.39%