香港浸會大學
HONG KONG BAPTIST UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE
HONG KONG BAPTIST UNIVERSITY
香港浸會大學計算機科學系

Prompt Distillation for Efficient LLM-based Recommendation

Lei Li[1], Yongfeng Zhang[2], Li Chen[1]

[1] Hong Kong Baptist University, [2] Rutgers University

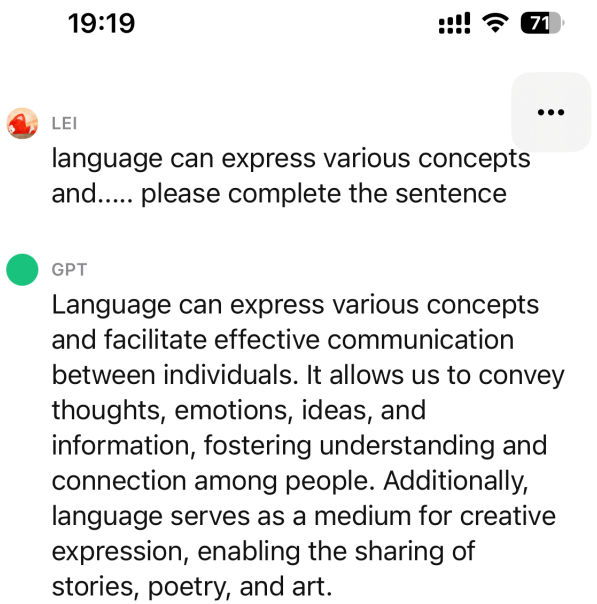
csleili@comp.hkbu.edu.hk

Oct. 25, 2023

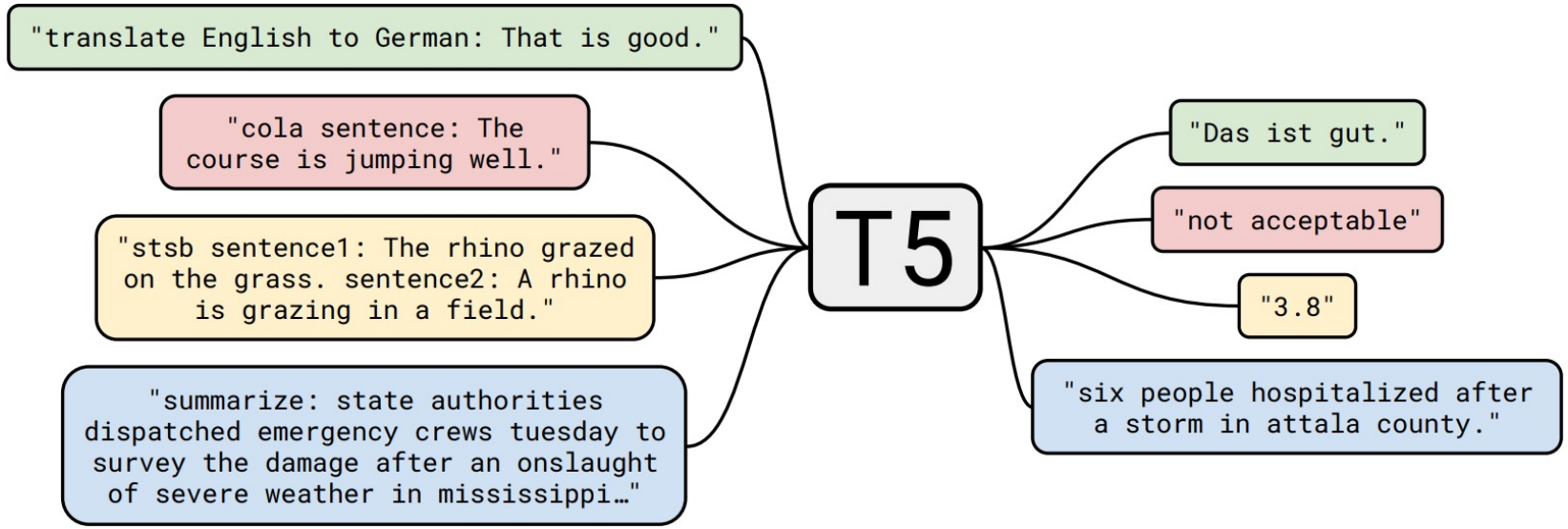
The 32nd ACM International Conference on Information and Knowledge Management (CIKM'23)

Large Language Models (LLM)

ChatGPT [1]



T5 [2]

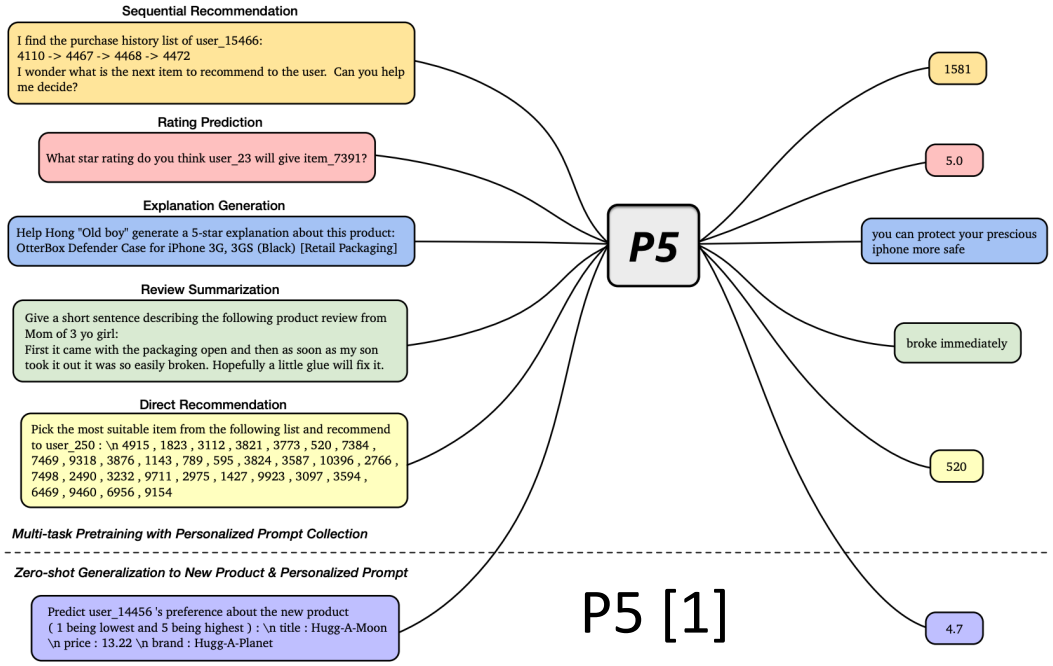


[1] https://openai.com/blog/chatgpt

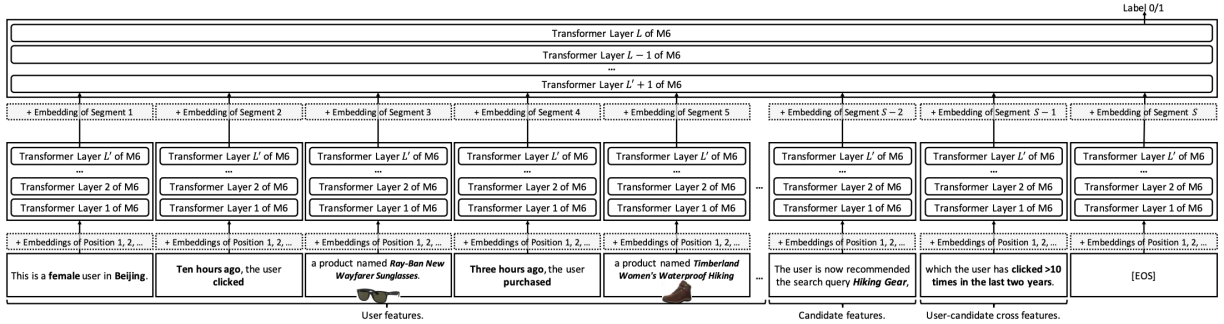
[2] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research (2020).

LLM-based Recommender Systems

- Tasks represented as a sequence-to-sequence problem
  - Users and items represented by IDs/metadata and filled in a template



P5 [1]



M6-Rec [2]

[1] Geng, Shijie, et al. "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys'22.

[2] Cui, Zeyu, et al. "M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv'22.

Discrete Prompt

- Adapt different tasks to LLM
- Require human labor to design templates

Task	Input (X)	Template	Output (Y)
Sentiment Classification	I love this book.	X The book is Y	great boring ...
Text Summarization	The Omicron ...	X TL;DR: Y	COVID-19 ... Pandemic ... ...
Machine Translation	Elle m'a apprivoisé.[3]	French: X English: Y	She tamed ... The flower ... ...

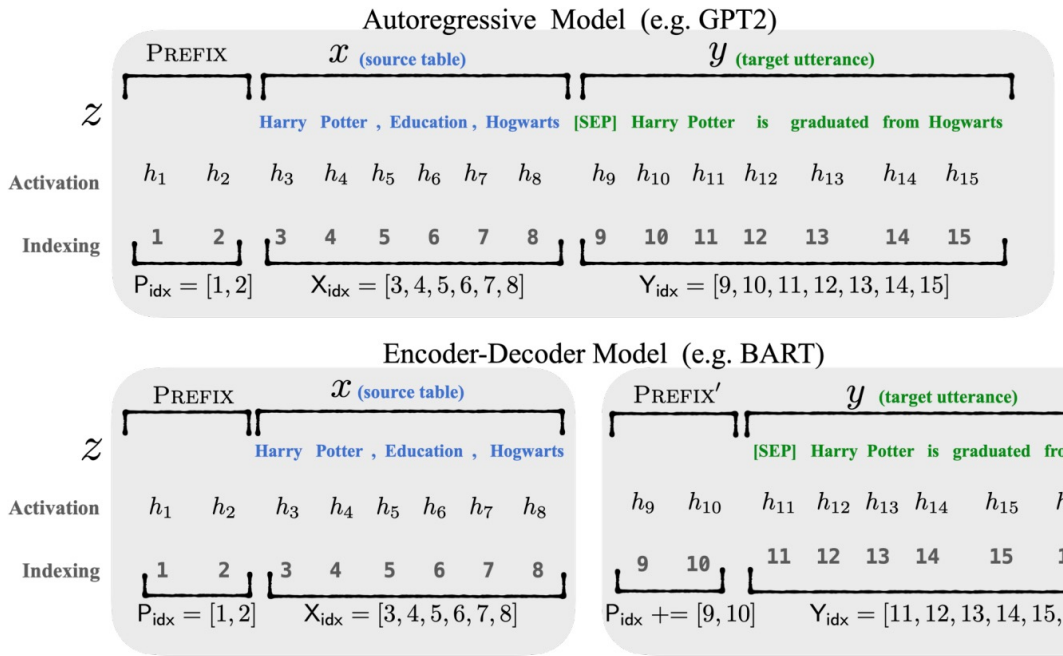
[1] Li, Lei, et al. "Personalized prompt learning for explainable recommendation." ACM Transactions on Information Systems (2023).

Problems with Discrete Prompt

- Long prompt takes time to process, and thus could be less efficient
- The key information to recommendation models is the user and item IDs, so the discrete prompt could be a little noisy
- Extensive fine-tuning is needed to bridge the gap between IDs and prompt words
  - Word embeddings capture the relation between words
  - ID embeddings encode users' preferences towards items

Continuous Prompt

- Prompt does not need to be text
- Continuous prompt vectors could be more expressive
  - They do not map to any words



Summarization Example

Article: Scientists at University College London discovered people tend to think that their hands are wider and their fingers are shorter than they truly are.They say the confusion may lie in the way the brain receives information from different parts of the body.Distorted perception may dominate in some people, leading to body image problems ... [ignoring 308 words] could be very motivating for people with eating disorders to know that there was a biological explanation for their experiences, rather than feeling it was their fault."

Summary: The brain naturally distorts body image - a finding which could explain eating disorders like anorexia, say experts.

Table-to-text Example

Table: name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

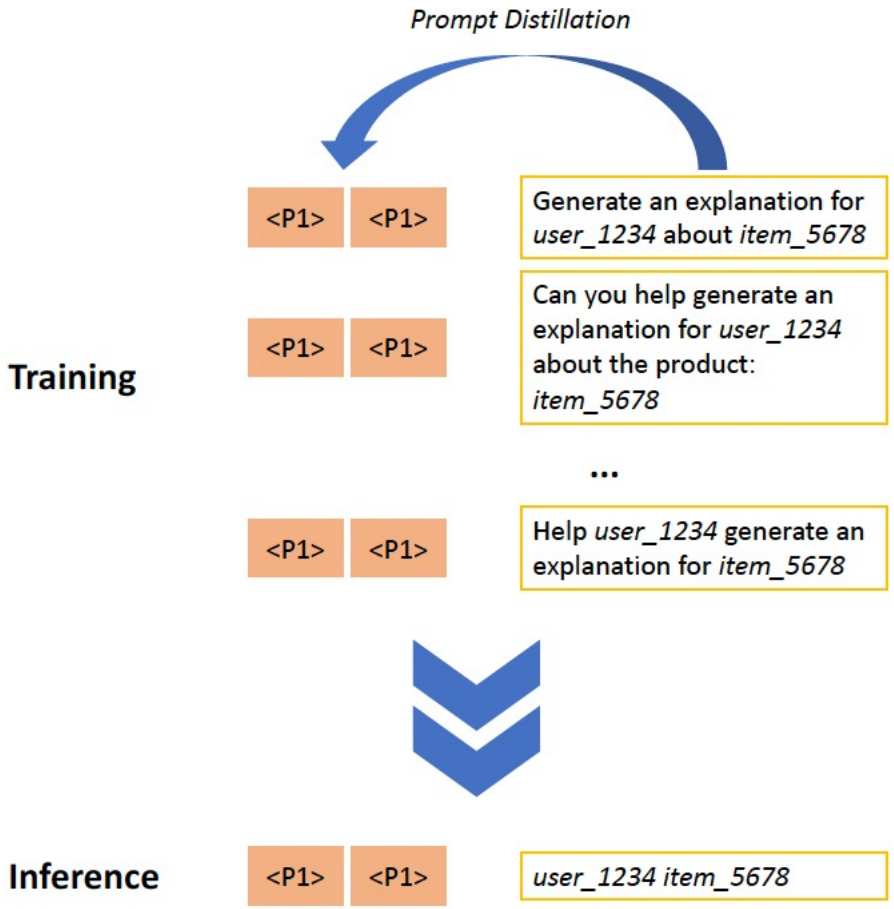
Textual Description: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

[1] Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." ACL'21.

Prompt Distillation

- To bridge the ID-word gap and reduce inference time
  - Problems of discrete prompt
- Assign each task a set of vectors

Definition (Prompt Distillation): *We call an approach prompt distillation if it can shorten a long prompt without sacrificing an LLM's performance on the testing tasks. The distilled short prompt can either be free text or vectors.*

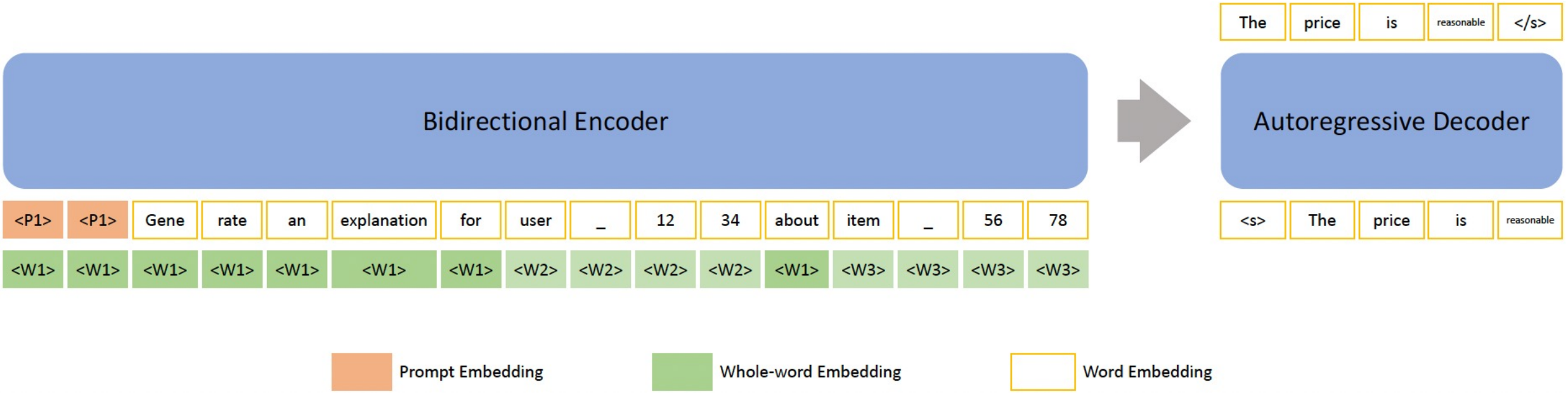


Tasks Formulation

- Explanation Generation
  - Generate an explanation to justify why an item is recommended to a user
- Top-N Recommendation
  - Predict N items that a user might be interested in based on the items that the user interacted with
- Sequential Recommendation
  - Predict the next item that a user is likely to interact with based on the sequentially ordered items in the user's interaction history

Model Overview

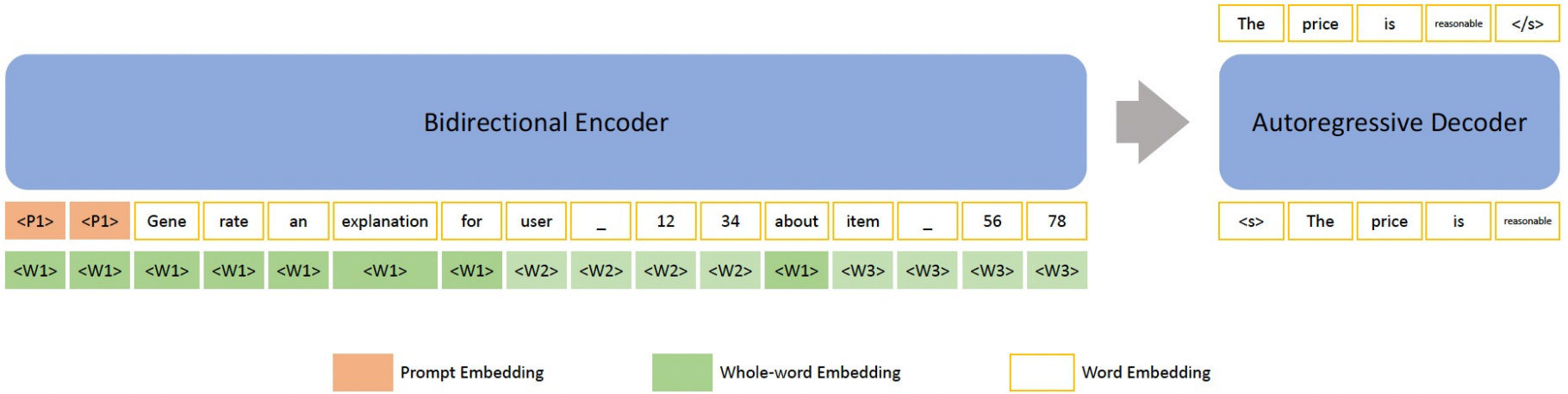
- Encoder-decoder structure based on T5 [1]
  - Encoder processes input information
  - Decoder performs autoregressive generation



[1] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research (2020).

Implementation Details

- Input is comprised of
  - Continuous prompt vectors
  - Discrete prompt template
  - IDs represented as strings
- Whole-word embedding [1] is applied to connect tokens of each ID string

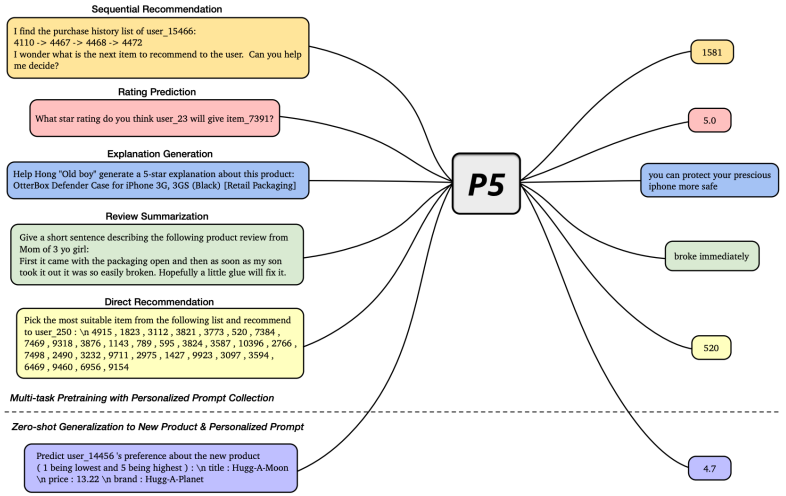


[1] Geng, Shijie, et al. "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys'22.

Training Strategy

- Each sample of three tasks is an input-output sentence pair (X, Y)
- The encoder takes in input (X) and the decoder fits output (Y)
- Negative log-likelihood (NLL) is adopted as loss function
- Samples of different tasks are mixed in one batch for training in P5 [1]

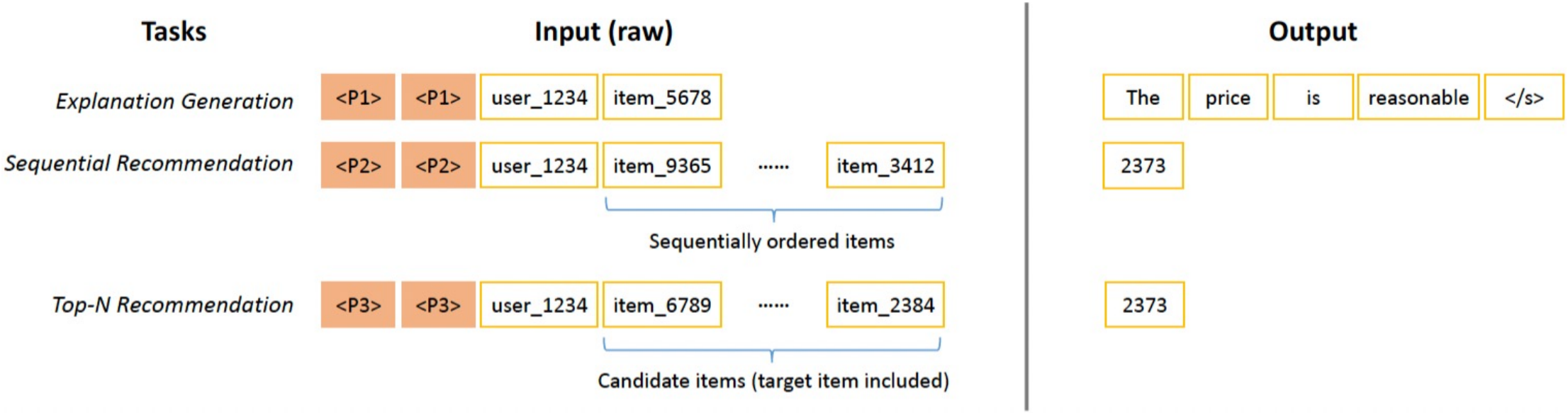
$$\mathcal{L}_\Theta = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \frac{1}{|Y|} \sum_{t=1}^{|Y|} -\log p(y_t | Y_{<t}, X)$$



[1] Geng, Shijie, et al. "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)." RecSys'22.

Efficiency Problem with Sample-mixed Training

- Input/output length of different tasks could vary
- Memory wasted on padding, small batch size, more iterations, increased training time



Task-alternated Training

- Alternately train the model with samples from the same task
- They generally have the same length
  - Less memory on padding
  - Larger batch size
  - Improved efficiency

Algorithm 1 Task-alternated Training

Input: Explanation set $\mathcal{E} = \{(u, i, E_{u,i})\}$, user set $\mathcal{U}$, prompt template sets for explanation $\mathcal{P}_e$, sequential recommendation $\mathcal{P}_s$ and top-N recommendation $\mathcal{P}_t$, number of negative items n

Output: Model parameters Θ

1: **repeat**
2: Uniformly draw a batch $\mathcal{B}$ from $\mathcal{E}$ // explanation
3: **for** $(u, i, E_{u,i})$ in $\mathcal{B}$ **do**
4: Draw a template $prompt(\cdot)$ from $\mathcal{P}_e$
5: $x \leftarrow prompt(u, i), y \leftarrow E_{u,i}$
6: **end for**
7: $X \leftarrow [x_1, ..., x_{|\mathcal{B}|}], Y \leftarrow [y_1, ..., y_{|\mathcal{B}|}]$
8: Update Θ with $\mathcal{L}_\Theta$ in Eq. (1) by feeding (X, Y)
9: Uniformly draw a batch $\mathcal{B}$ from $\mathcal{U}$ // sequential
10: **for** u in $\mathcal{B}$ **do**
11: Draw a template $prompt(\cdot)$ from $\mathcal{P}_s$, a segment $\widetilde{I^u}$ from $I^u_{|I^u|-2}$ // last two items for validation and testing
12: $x \leftarrow prompt(u, \widetilde{I^u_{|I^u|-1}}), y \leftarrow \widetilde{i^u_{|I^u|}}$
13: **end for**
14: Execute line 7 and 8
15: Uniformly draw a batch $\mathcal{B}$ from $\mathcal{U}$ // top-N
16: **for** u in $\mathcal{B}$ **do**
17: Draw a template $prompt(\cdot)$ from $\mathcal{P}_t$, an item i from $I^u_{|I^u|-2}$, n negative items $\mathcal{I}^n$ from $\mathcal{I}/\mathcal{I}^u$
18: Add i to $\mathcal{I}^n$ and shuffle
19: $x \leftarrow prompt(u, \mathcal{I}^n), y \leftarrow i$
20: **end for**
21: Execute line 7 and 8
22: **until** Convergence

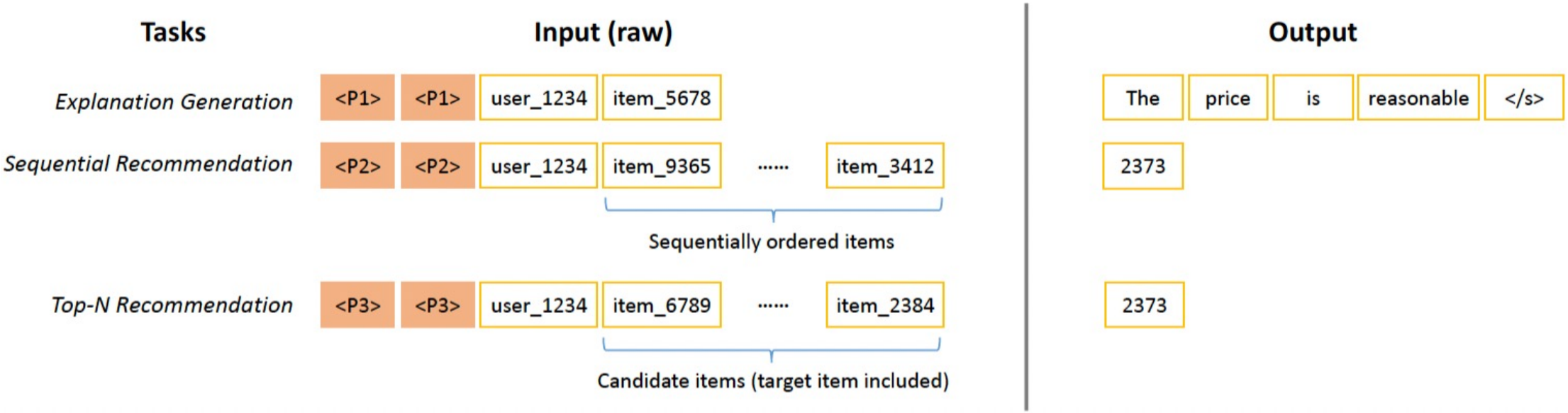
Explanation

Sequential

Top-N

Inference Stage

- Discrete prompt is removed so as to improve inference efficiency
- Beam search applied
  - Produce multiple generations, i.e., multiple recommendations for each user



Datasets

- Three amazon datasets
  - Sports
  - Beauty
  - Toys
- An explanation is a sentence mined from user reviews [1]
- Training : validation : test = 8:1:1

amazon

Dataset	Sports	Beauty	Toys
#Users	35,598	22,363	19,412
#Items	18,357	12,101	11,924
#Reviews	296,337	198,502	167,597
#Sparsity (%)	0.0453	0.0734	0.0724

[1] Li, Lei, et al. "Generate neural template explanations for recommendation." CIKM'20.

Sequential Recommendation Performance

- Outperform state-of-the-art baselines by a large margin

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^3-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	0.0272	0.0169	0.0361	0.0198	0.0503	0.0370	0.0659	0.0421	0.0648	0.0567	0.0709	0.0587
POD	**0.0496**	**0.0396**	**0.0576**	**0.0419**	**0.0537**	**0.0395**	**0.0688**	**0.0443**	**0.0691**	**0.0599**	**0.0742**	**0.0610**
Improvement (%)	82.35	134.32	49.61	105.39	6.76	6.76	4.40	5.23	6.64	5.64	4.65	3.92

Top-N Recommendation Performance

- Outperform classic recommendation baselines
- Top-1 recommendation performance is quite good
  - Great practical value in real-world systems, e.g., conversational recommendation
  - System can only display a few or just one 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
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
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
P5	0.0567	0.1514	0.1049	0.2196	0.1269	0.0571	0.1566	0.1078	0.2317	0.1318	0.0451	0.1322	0.0889	0.2023	0.1114
POD	**0.0895**	**0.2086**	**0.1506**	**0.2873**	**0.1756**	**0.0829**	**0.1926**	**0.1391**	**0.2670**	**0.1629**	**0.0567**	**0.1433**	**0.1009**	**0.2082**	**0.1215**
Improvement (%)	57.85	37.24	43.57	7.56	35.49	45.18	22.99	29.04	3.77	23.60	25.72	8.40	13.50	0.24	9.07

Explanation Generation Performance

- Comparable performance to baseline methods
- BLEU and ROUGE overly stress the matching between generation and ground-truth [1]
  - Put LLM that can generate expressive content at disadvantage

Methods	Sports				Beauty				Toys			
	BLUE-4	ROUGE-1	ROUGE-2	ROUGE-L	BLUE-4	ROUGE-1	ROUGE-2	ROUGE-L	BLUE-4	ROUGE-1	ROUGE-2	ROUGE-L
Att2Seq	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	**1.1541**	14.8497	**2.1413**	**11.4143**	1.9861	**14.2716**	3.6718	**11.7010**
POD	**1.0013**	**14.0168**	**2.0436**	**11.1236**	1.0630	**15.2517**	1.5737	11.3283	**2.3053**	12.2889	**3.8512**	10.3923
Improvement (%)	40.79	8.70	53.85	12.78	-7.89	2.71	-26.51	-0.75	16.07	-13.89	4.89	-11.18

[1] Wang, Xiaolei, et al. "Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models." EMNLP'23.

Training Efficiency

- Largely reduce training time
  - Overall time
  - Time per epoch

Training Strategies	Time	Epochs	Time/Epoch
Sample-mixed	15h59m	13	1h14m
Task-alternated	6h55m	22	19m
Improvement (%)	56.73	-	74.32

Inference Efficiency

- Slightly improve the inference efficiency without sacrificing much accuracy after removing discrete prompt
- More work can be done
  - Light-weighted model

Methods	Sequential Recommendation					Top-N Recommendation					Explanation Generation (%)			
	HR@5	NDCG@5	HR@10	NDCG@10	Time	HR@5	NDCG@5	HR@10	NDCG@10	Time	BLEU-4	ROUGE-2	ROUGE-L	Time
Continuous+Discrete	0.0509	0.0411	0.0583	0.0432	24m6s	0.2079	0.1508	0.2882	0.1763	48m31s	1.0012	2.0436	11.1202	9m30s
Continuous only (POD)	0.0496	0.0396	0.0576	0.0419	22m17s	0.2086	0.1506	0.2873	0.1756	47m13s	1.0013	2.0436	11.1236	8m59s
Improvement (%)	-2.55	-3.65	-1.20	-3.01	7.54	0.34	-0.13	-0.31	-0.40	2.68	0.01	0.00	0.03	5.44

Conclusion

- Present a simple but effective PrOmpt Distillation (POD) approach
  - To distill the knowledge of discrete prompt templates into continuous prompt vectors for LLM-based recommendation models
- Propose a Task-alternated Training strategy
  - To improve the efficiency of training an LLM-based recommendation model

- Future Work
  - Explore cross-task prompt transfer to generalize LLM to new recommendation tasks
  - Improve inference efficiency of LLM for recommender systems

ACM TORS Special Issue Call for Papers

- **Topic: Large Language Models for Recommender Systems**
- Journal: ACM Transactions on 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

Official webpage

Q&A

Thank you!

csleili@comp.hkbu.edu.hk

lileipisces.github.io