PERSONALIZED PROMPT LEARNING FOR EXPLAINABLE RECOMMENDATION

STUDY GROUP

THUIR

任一 renyi1006@gmail.com

2022/09/15

MINI HALL, FIT 1-506, TSINGHUA UNIVERSITY

0A ABOUT

Personalized Prompt Learning for Explainable Recommendation

LEI LI, Hong Kong Baptist University, China
YONGFENG ZHANG, Rutgers University, USA
LI CHEN, Hong Kong Baptist University, China

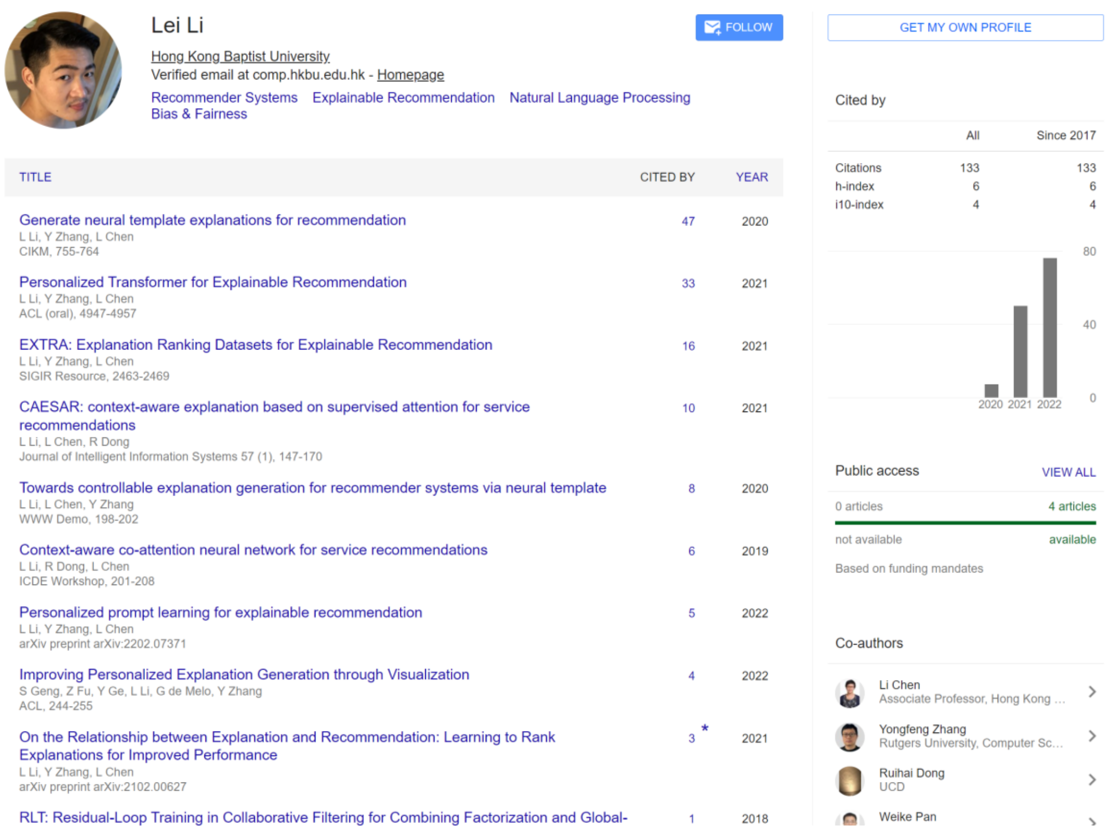
Personalized prompt learning for explainable recommendation

L Li, Y Zhang, L Chen - arXiv preprint arXiv:2202.07371, 2022 - arxiv.org

Providing user-understandable explanations to justify recommendations could help users better understand the recommended items, increase the system's ease of use, and gain users' trust. A typical approach to realize it is natural language generation. However, previous works mostly adopt recurrent neural networks to meet the ends, leaving the potentially more effective pre-trained Transformer models under-explored. In fact, user and item IDs, as important identifiers in recommender systems, are inherently in different …

☆ 保存 ㄗㄗ 引用 被引用次数：5 相关文章 所有 2 个版本 »

0A ABOUT


Lei Li

FOLLOW

Hong Kong Baptist University
Verified email at comp.hkbu.edu.hk - Homepage
Recommender Systems Explainable Recommendation Natural Language Processing Bias & Fairness

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Generate neural template explanations for recommendation L Li, Y Zhang, L Chen CIKM, 755-764	47	2020
Personalized Transformer for Explainable Recommendation L Li, Y Zhang, L Chen ACL (oral), 4947-4957	33	2021
EXTRA: Explanation Ranking Datasets for Explainable Recommendation L Li, Y Zhang, L Chen SIGIR Resource, 2463-2469	16	2021
CAESAR: context-aware explanation based on supervised attention for service recommendations L Li, L Chen, R Dong Journal of Intelligent Information Systems 57 (1), 147-170	10	2021
Towards controllable explanation generation for recommender systems via neural template L Li, L Chen, Y Zhang WWW Demo, 198-202	8	2020
Context-aware co-attention neural network for service recommendations L Li, R Dong, L Chen ICDE Workshop, 201-208	6	2019
Personalized prompt learning for explainable recommendation L Li, Y Zhang, L Chen arXiv preprint arXiv:2202.07371	5	2022
Improving Personalized Explanation Generation through Visualization S Geng, Z Fu, Y Ge, L Li, G de Melo, Y Zhang ACL, 244-255	4	2022
On the Relationship between Explanation and Recommendation: Learning to Rank Explanations for Improved Performance L Li, Y Zhang, L Chen arXiv preprint arXiv:2102.00627	3 *	2021
RLT: Residual-Loop Training in Collaborative Filtering for Combining Factorization and Global-	1	2018

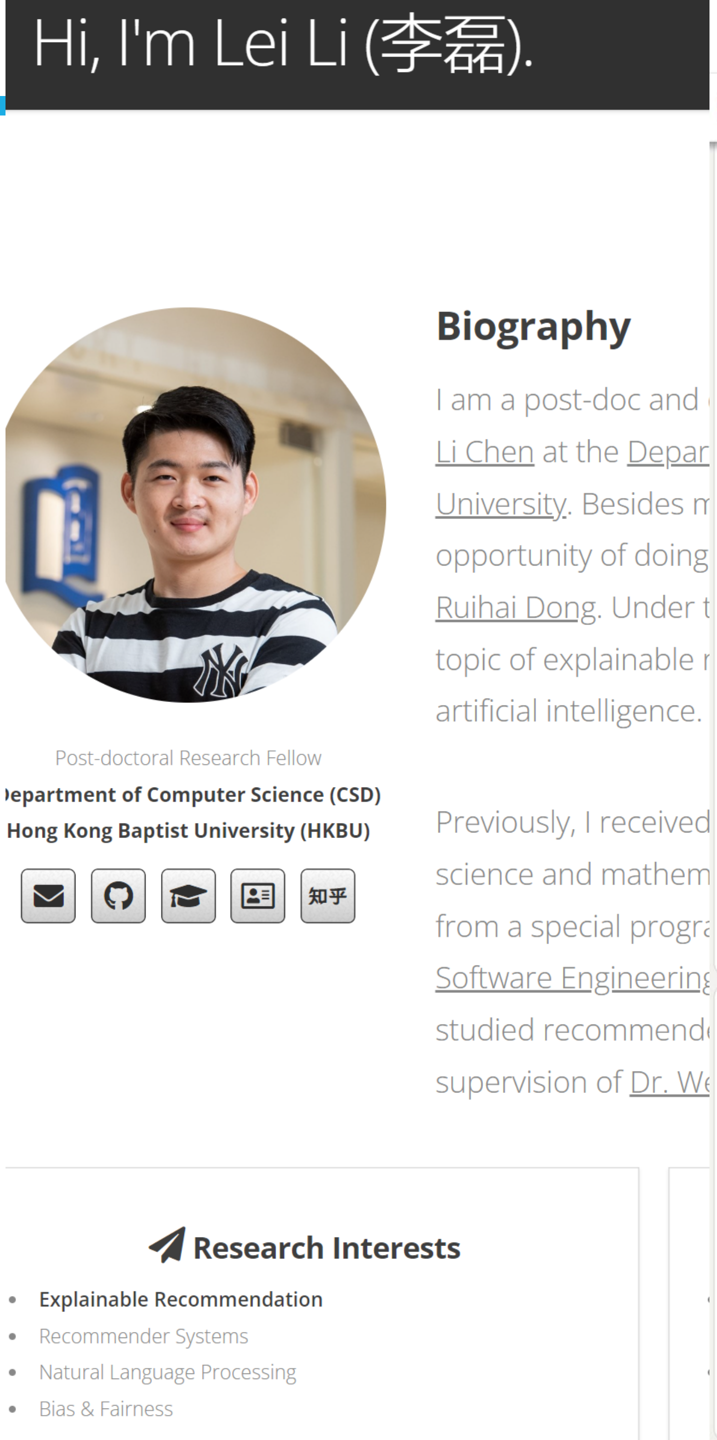
Cited by

	All	Since 2017
Citations	133	133
h-index	6	6
i10-index	4	4

80 40 0
2020 2021 2022

Public access VIEW ALL
0 articles 4 articles
not available available
Based on funding mandates

Co-authors
Li Chen
Associate Professor, Hong Kong ...
Yongfeng Zhang
Rutgers University, Computer Sc...
Ruihai Dong
UCD
Weike Pan


Hi, I'm Lei Li (李磊).

Post-doctoral Research Fellow
Department of Computer Science (CSD)
Hong Kong Baptist University (HKBU)

知乎

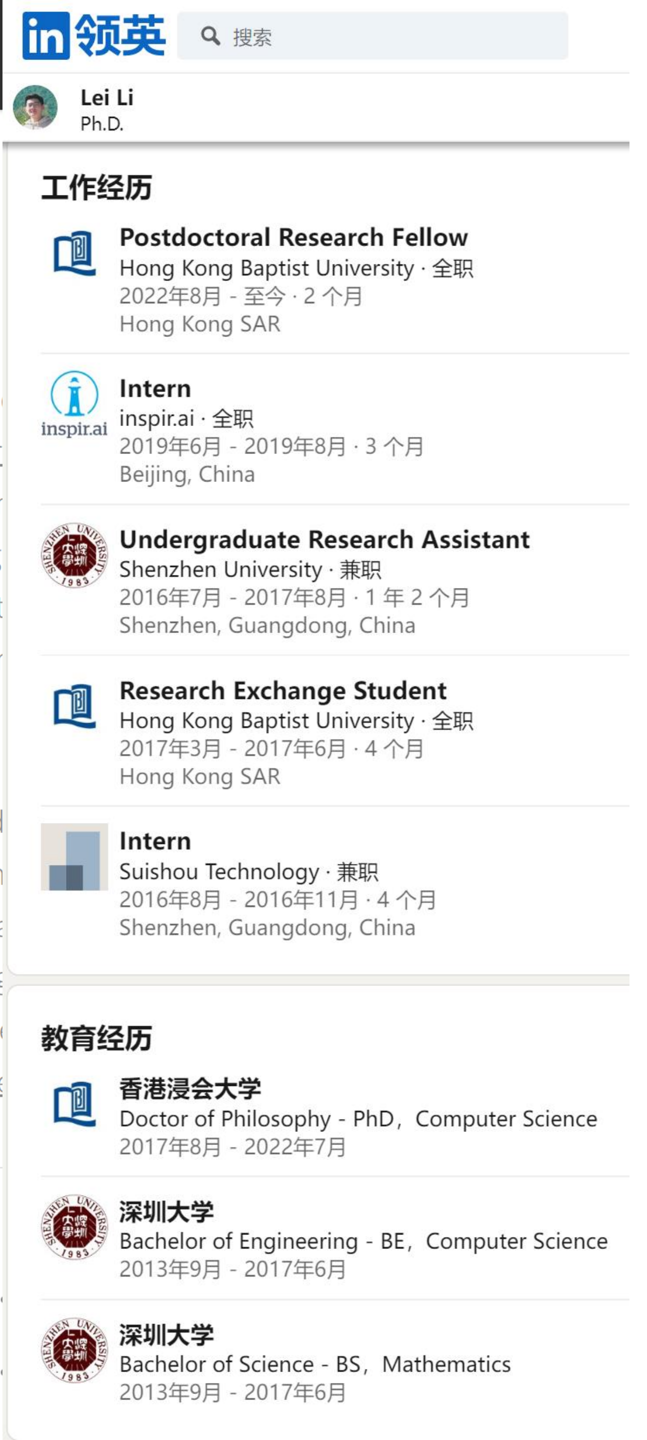
Biography

I am a post-doc and
Li Chen at the Depar
University. Besides m
opportunity of doing
Ruihai Dong. Under t
topic of explainable r
artificial intelligence.

Previously, I received
science and mathem
from a special progra
Software Engineering
studied recommend
supervision of Dr. We

Research Interests

- Explainable Recommendation
- Recommender Systems
- Natural Language Processing
- Bias & Fairness


in 领英 搜索

Lei Li
Ph.D.

工作经历

Postdoctoral Research Fellow
Hong Kong Baptist University · 全职
2022年8月 - 至今 · 2 个月
Hong Kong SAR

Intern
inspir.ai · 全职
2019年6月 - 2019年8月 · 3 个月
Beijing, China

Undergraduate Research Assistant
Shenzhen University · 兼职
2016年7月 - 2017年8月 · 1 年 2 个月
Shenzhen, Guangdong, China

Research Exchange Student
Hong Kong Baptist University · 全职
2017年3月 - 2017年6月 · 4 个月
Hong Kong SAR

Intern
Suishou Technology · 兼职
2016年8月 - 2016年11月 · 4 个月
Shenzhen, Guangdong, China

教育经历

香港浸会大学
Doctor of Philosophy - PhD，Computer Science
2017年8月 - 2022年7月

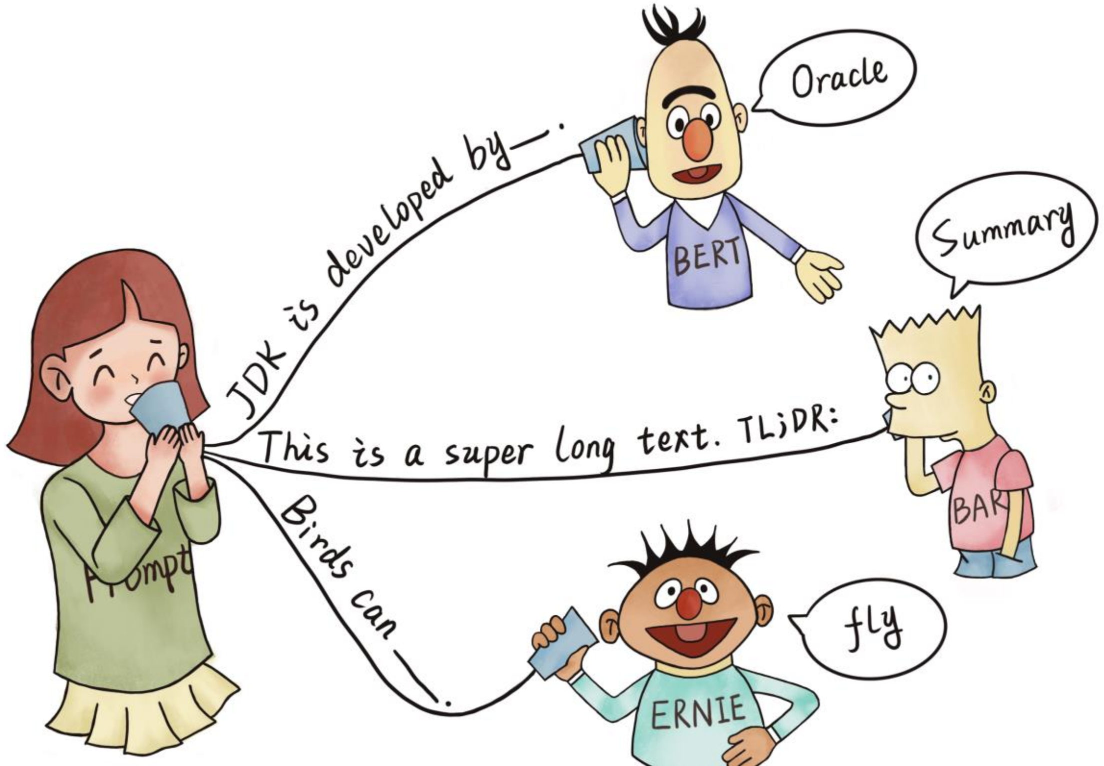
深圳大学
Bachelor of Engineering - BE，Computer Science
2013年9月 - 2017年6月

深圳大学
Bachelor of Science - BS，Mathematics
2013年9月 - 2017年6月

0B PRELIMINARIES

PROMPT LEARNING – WHAT

- Traditional supervised learning
  - Input: x
  - Label: y
  - Predicted output: P(y|x)
- Prompt-based learning
  - Based on language models
  - **Model the probability of text directly**



0B PRELIMINARIES

PROMPT LEARNING – WHAT

- Four paradigms in NLP
  - Fully Supervised Learning
    - Neural/Non-Neural Network
  - Pre-train, Fine-tune
  - Pre-train, Prompt, Predict

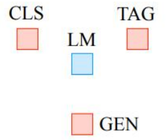
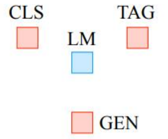
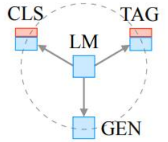
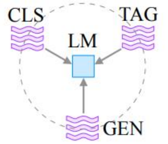
Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Table 1: Four paradigms in NLP. The "**engineering**" column represents the type of engineering to be done to build strong systems. The "**task relation**" column, shows the relationship between language models (LM) and other NLP tasks (CLS: classification, TAG: sequence tagging, GEN: text generation). : fully unsupervised training. : fully supervised training. : Supervised training combined with unsupervised training. indicates a textual prompt. Dashed lines suggest that different tasks can be connected by sharing parameters of pre-trained models. "LM→Task" represents *adapting LMs (objectives) to downstream tasks* while "Task→LM" denotes *adapting downstream tasks (formulations) to LMs*.

0B PRELIMINARIES

PROMPT LEARNING – HOW

- Four paradigms in NLP
  - Fully Supervised Learning
    - Neural/Non-Neural Network
  - Pre-train, Fine-tune
  - Pre-train, Prompt, Predict

Type	Task	Input ([X])	Template	Answer ([Z])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...

Table 3: Examples of *input*, *template*, and *answer* for different tasks. In the **Type** column, "CLS" is an abbreviation for "classification". In the **Task** column, "NLI" and "NER" are abbreviations for "natural language inference" (Bowman et al., 2015) and "named entity recognition" (Tjong Kim Sang and De Meulder, 2003) respectively.

0B PRELIMINARIES

PROMPT LEARNING – WHAT+

- Prompting Function
- Input
- Output
- Prompt
- Filled Prompt
- Answered Prompt
- Answer

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	[X] Overall, it was a [Z] movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot [Z] where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a [Z] movie.	A text where [X] is instantiated by input $\boldsymbol{x}$ but answer slot [Z] is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot [Z] is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot [Z] is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills [Z]

Table 2: Terminology and notation of prompting methods. $\boldsymbol{z}^*$ represents answers that correspond to true output $\boldsymbol{y}^*$.

0B PRELIMINARIES

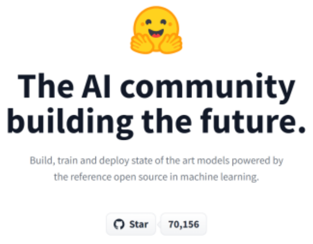
PROMPT LEARNING – WHY

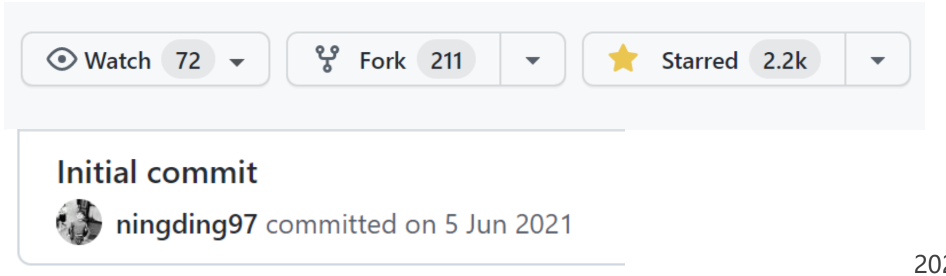
- Pros
  - Stand on the shoulders of **pre-trained models**, compared to fully supervised learning.
    - *If I have seen further it is by standing on the shoulders of giants. – Issac Newton, a letter to Robert Hooke in 1675*
  - No need to **define another objective to fine-tune**, which saves energy and seems more intuitive, compared to the pre-train, fine-tune paradigm.
    - Downstream tasks are also text generation in prompt learning, **same as** what the model do in the pre-train phase.
  - Suitable for **a wide range of tasks**, given appropriate prompts.
    - Possible for **few-shot and zero-shot learning**.
      - Converting various problems into modeling text probability, casting light on downstream tasks with few or no labels.
    - Template words are not necessarily composed of natural language tokens
      - They could be virtual words (e.g. represented by **numeric ids**) which would be embedded in a continuous space later
      - Some prompting methods even generate continuous vectors directly (more in §4.3.2).
- Cons
  - Needs **prompt engineering**.
  - **Hallucination**. May lead to **biased** and offensive outputs.

0B PRELIMINARIES

PROMPT LEARNING – WHO, WHEN, WHERE

- Pilot Works
  - Language Models are Few-Shot Learners, NeurIPS 2020, Tom Brown (Co-Founder at Anthropic), #cite:5239
  - Exploring the Limits of **Transfer Learning** with a Unified Text-to-Text Transformer, JMLR 2019, Colin Raffel (AP, UNC/Faculty Researcher, Hugging Face), #cite:4369
  - Language Models as Knowledge Bases?, EMNLP 2019, Research Tech Lead Manager at the FAIR (Fundamental AI Research) team of Meta AI, #cite:852
- Main Reference for This Section
  - **Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing,** Pengfei Liu (postdoc at the Language Technologies Institute (LTI) of Carnegie Mellon University, working with Prof. Graham Neubig) , arXiv preprint arXiv:2107.13586, 2021.
- *Must-read papers on prompt-based tuning for pre-trained language models, PromptPapers,* THUNLP 2021 https://github.com/thunlp/PromptPapers

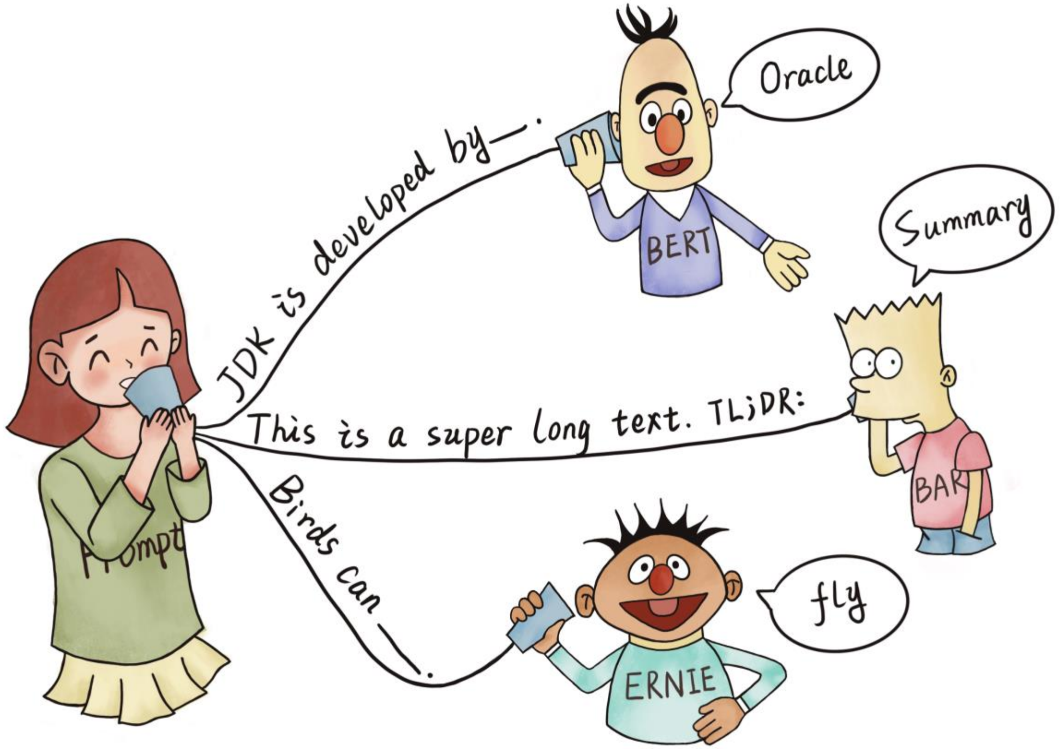




0B PRELIMINARIES

PROMPT LEARNING – TAKEAWAY

- Prompt-based learning
  - Based on language models
  - **Model the probability of text directly**
  - Rather than model P(y|x) as traditional supervised learning
  - Pros: pre-train power + wide application + little cost
  - Cons: prompt engineering, hallucination



1 MOTIVATION

BACKGROUND

- **Explanations** for recommendation results are important! There's a variety of explanation **style**.
  - Pre-defined templates, highlighted image regions, automatically generated sentences
  - Automatically generated sentences
    - Availability of **textual data** on online commercial platforms(Amazon, Yelp, JD, Taobao) like user **reviews**
    - Advancement of **natural language generation** (NLG) techniques like RNN, Transformer, PLM (pre-trained language models)
- PLMs, though powerful, are **nearly impossible to do customized modifications**, due to massive parameters and training data
  - GPT-3 has 175 billion parameters and needs 570GB of textual data to train!
- Fortunately, **PROMPT LEARNING** comes to rescue!
  - Prompt learning aims to **DIRECTLY** **model text probability** of the converted downstream task, instead of fine-tuning for a different downstream task, which saves energy and sounds more intuitive.
- How to **apply prompt learning to recommendation explanation generation** is the main concern of this paper. (**FIRST** to introduce prompt learning to the community of recommender systems)

2 RELATED WORK

2.1 EXPLAINABLE RECOMMENDATION

- Two major perspectives on explainable recommendation
  - Human-computer interaction
    - How people perceive different styles of explanation
  - Machine learning
    - Various types of explanation style,
      - Pre-defined templates, item features, ranked text, image visualizations, knowledge graph paths, reasoning rules
      - Generating natural language explanations
        - Previous work mostly rely on RNN, unpretrained Transformer
        - **A chance for pre-trained models!**

2 RELATED WORK

2.2 TRANSFORMER AND PRE-TRAINED MODELS

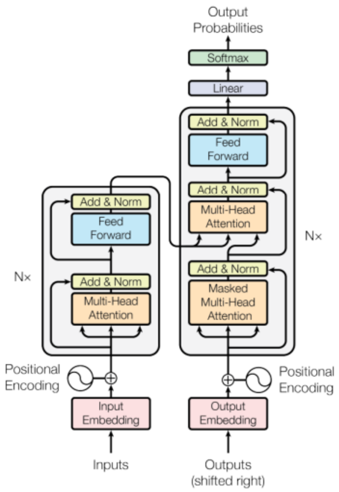


Figure 1: The Transformer - model architecture.

- Transformer
  - First brought to **machine translation** with encoder-decoder (Attention is all you need, NIPS’17)
  - More **powerful** with pre-training+fine-tuning paradigm, confirmed effective on a wide range of natural language understanding tasks (commonsense reasoning and question answering)
  - Prompt learning makes it **possible** to transfer PLMs to other downstream tasks (may differ from pre-trained tasks) with **few-shot or even zero-shot** learning standing, on the shoulder of **powerful text modeling ability** of PLMs.
  - Prompt learning has been successfully applied to a lot of research or application fields, such as domain adaptation, text summarization and image captioning.
    - However, the application of **prompt learning on recommender systems is still unexplored**! **A VIRGIN TERRITORY!**

But why, some say, the Moon? Why choose this as our goal? And they may well ask, why climb the highest mountain? Why, 35 years ago, fly the Atlantic? Why does Rice play Texas?

We choose to go to the Moon. *We choose to go to the Moon... We choose to go to the Moon in this decade and do the other things,* ***not because they are easy, but because they are hard****; because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we* ***are willing to accept, one we are unwilling to postpone, and one we intend to win, and the others****, too.*[11] *-- USA* President John F. Kennedy speaking at Rice University on September 12, 1962

2 RELATED WORK

2.3 PERSONALIZED NATURAL LANGUAGE GENERATION

- Personalization of natural language generations is important, in explainable recommendation, review summarization and dialog systems.
  - User and item IDs are important identifiers for personalization.
    - Previous methods like MLP encodes ID into a context vector, from which **RNN decodes** into a word sequence.
      - Review generation, tip generation, explanation generation…
    - Not suitable for pre-trained models that were already trained on a massive amount of raw text
      - Probably **NO PROPER SOLUTION** for heterogeneous generation (i.e. IDs and words).
        - P5/T5?
    - Previous works with Transformer or PLMs for personalized NLG replace IDs with text segments, such as persona attributes, movie titles and item features (somewhat similar to prompt learning?)
  - This paper further investigates how to **incorporate continuous prompts (i.e. ID vectors) into pre-trained models**, in order to retain much information as possible with the potential power of continuous prompts.

3 METHODOLOGY

3.0 GOAL

- Generate a natural language sentence $\hat{E}_{u,i}$ for a given user-item pair (u, i) to justify **why** i is recommended to u
  - i could be predicted for u by a recommendation model (e.g. MF) or resulted from her/his true behaviour
  - Only u and i are used as input for producing the explanation, making it **compatible with any recommendation model** where user and item IDs are indispensable.

3 METHODOLOGY

3.1 TRANSFORMER, PRE-TRAINED LANGUAGE MODELS AND PROMPT LEARNING

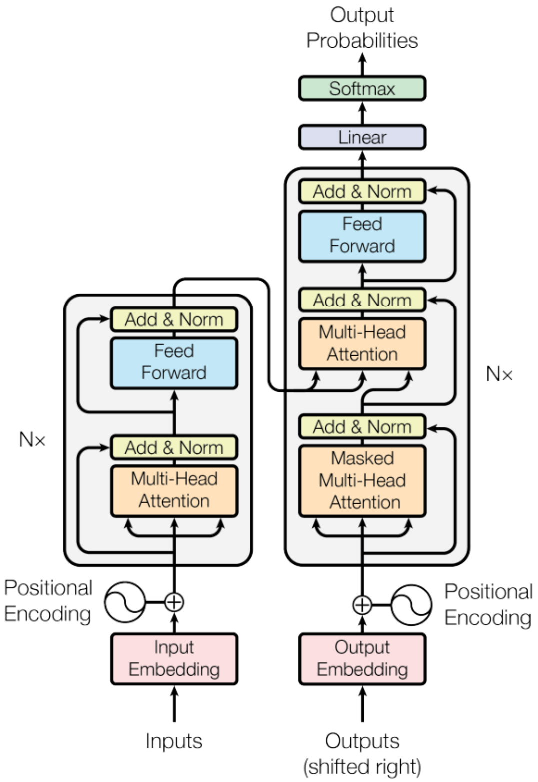


Figure 1: The Transformer - model architecture.

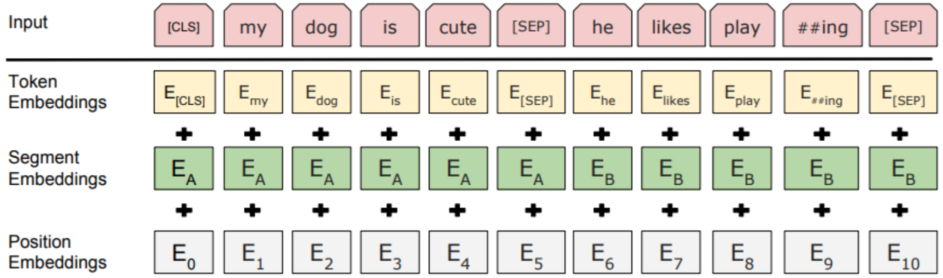


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

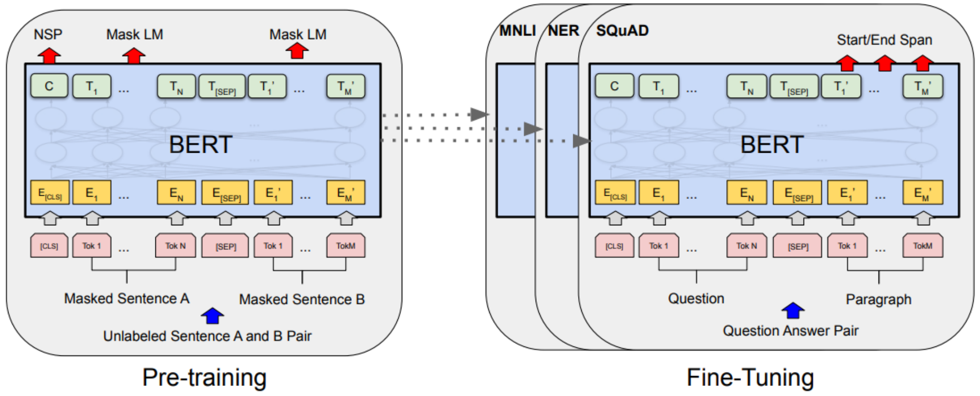


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Table 2. Prompt learning for typical natural language processing tasks [35]. In our explanation generation task, the template words "Explain the recommendation:" are removed.

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.[2]	French: X English: Y	She tamed ... The flower ... ...
Explanation Generation	room location ... X1: user123abc X2: item456def	X (Explain the recommendation:) Y X1 X2 (Explain the recommendation:) Y	The room ... The breakfast ... The location ... ...

3 METHODOLOGY

3.2 DISCRETE PROMPT LEARNING

- PLMs were trained on a large amount of word tokens, which are inherently in a different semantic space as ID tokens.
  - But IDs (user/item IDs) are indispensable in recommender systems.
  - Alternative –find some word tokens to represent the IDs
    - Move titles
    - **Item features –PEPLER-D** (**PE**rsonalized **P**rompt **L**earning for **E**xplainable **R**ecommendation, **D**iscrete Form)

3 METHODOLOGY

3.2 DISCRETE PROMPT LEARNING –ITEM FEATURE AS DISCRETE PROMPT

- PEPLER-D (**PE**rsonalized **P**rompt **L**earning for **E**xplainable **R**ecommendation, **D**iscrete Form)
  - Given a user u (or an item i), we can obtain all the associated item features F_u (or F_i) appeared in training set.
  - For each user-item pair (u, i), we categorize these features into two groups: the intersection of F_u and F_i (i.e., $F_u \cap F_i$), and the remaining features $(F_u \cup F_i) \setminus (F_u \cap F_i)$
  - Then, the discrete prompt for this user-item pair is defined as $F_{u,i} = [(F_u \cap F_i), (F_u \cup F_i) \setminus (F_u \cap F_i)]$

To prevent from too many features, we may chop off some less informative ones at the right-hand side.

During the training stage, the input sequence to the pre-trained model can be represented as $S = [f_1, \cdots, f_{|F_{u,i}|}, e_1, \cdots, e_{|E_{u,i}|}]$, where $f_1, \cdots, f_{|F_{u,i}|}$ are the discrete prompt consisting of features, $e_1, \cdots, e_{|E_{u,i}|}$ are the explanation's word sequence, and $|F_{u,i}|$ and $|E_{u,i}|$ respectively denote the number of features and explanation words. Because all the tokens in sequence S are of the same type, i.e., words, we can perform embedding look-up once for them all, which gives the sequence's token representation $[\mathbf{f}_1, \cdots, \mathbf{f}_{|F_{u,i}|}, \mathbf{e}_1, \cdots, \mathbf{e}_{|E_{u,i}|}]$. The input representation of the sequence to the model is the addition of the token representation, and the positional representation $[\mathbf{p}_1, \cdots, \mathbf{p}_{|S|}]$ that encodes the position of each token in the sequence. We denote the input representation as $\mathbf{S}_0 = [\mathbf{s}_{0,1}, \cdots, \mathbf{s}_{0,|S|}]$, where $|S|$ is the length of the sequence.

3 METHODOLOGY

3.2 DISCRETE PROMPT LEARNING –ITEM FEATURE AS DISCRETE PROMPT

- PEPLER-D (**PE**rsonalized **P**rompt **L**earning for **E**xplainable **R**ecommendation, **D**iscrete Form)
  - Given a user u (or an item i), we can obtain all the associated item features F_u (or F_i) appeared in training set.
  - For each user-item pair (u, i), we categorize these features into two groups: the intersection of F_u and F_i (i.e., $F_u \cap F_i$), and the remaining features $(F_u \cup F_i) \setminus (F_u \cap F_i)$
  - Then, the discrete prompt for this user-item pair is defined as $F_{u,i} = [(F_u \cap F_i), (F_u \cup F_i) \setminus (F_u \cap F_i)]$

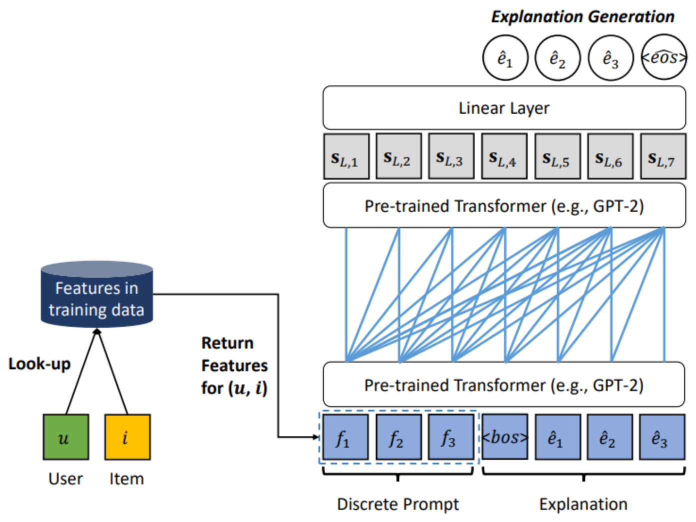


Fig. 3. Our proposed method PEPLER-D that utilizes item features as discrete prompt for explanation generation.

3 METHODOLOGY

3.2 DISCRETE PROMPT LEARNING –ITEM FEATURE AS DISCRETE PROMPT

After passing S_0 through pre-trained Transformer, we obtain the sequence's final representation $\mathbf{S}_L = [\mathbf{s}_{L,1}, \cdots, \mathbf{s}_{L,|S|}]$. Then, we apply a linear layer to each token's final representation to map it onto a $|\mathcal{V}|$-sized vector. As an example, $\mathbf{s}_{L,t}$ becomes $\mathbf{c}_t$ after passing through this layer:

$$\mathbf{c}_t = \operatorname{softmax}(\mathbf{W}^v \mathbf{s}_{L,t} + \mathbf{b}^v) \tag{5}$$

where $\mathbf{W}^v \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{b}^v \in \mathbb{R}^{|\mathcal{V}|}$ are weight parameters, and softmax$(\cdot)$ is the softmax function. The vector $\mathbf{c}_t$ represents the probability distribution over the vocabulary $\mathcal{V}$. For model learning, we adopt negative log-likelihood (NLL) as the loss function, and compute the mean of user-item pairs in the training set:

$$\mathcal{L}_D = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} -\log c_{|F_{u,i}|+t}^{e_t} \tag{6}$$

where the probability $c_t^{e_t}$ is offset by $|F_{u,i}|$ positions because the explanation is placed at the end of the sequence.

3 METHODOLOGY

3.3 CONTINUOUS PROMPT LEARNING

- Conversion from IDs to words (i.e., features) may **lose information** about ID
  - Nearly impossible to convert the features back into IDs
  - Not elegant enough as it is in discrete prompt learning
- Prompts DO NOT necessarily be words or even human-readable!
  - **Vector representations**, either produced by other models or randomly initialized, **are enough** for our **computer** darling to work with
  - This type of human-incomprehensible prompts are formally termed ***continuous/soft prompt***
  - **ID vectors** could also be **DIRECTLY** used as continuous prompts to generate recommendation explanations
- Comments
  - Human may care about the comprehensibility of prompt words, while computer only care about numbers, i.e. vectors
  - Continuous prompt learning seems a more **intuitive** approach in recommender systems, where items are first represented by ID vectors by custom, matching continuous learning very well. There seems no necessity to use feature vectors to represent items just to cater for discrete prompt learning.

3 METHODOLOGY

3.3 CONTINUOUS PROMPT LEARNING –PEPLER

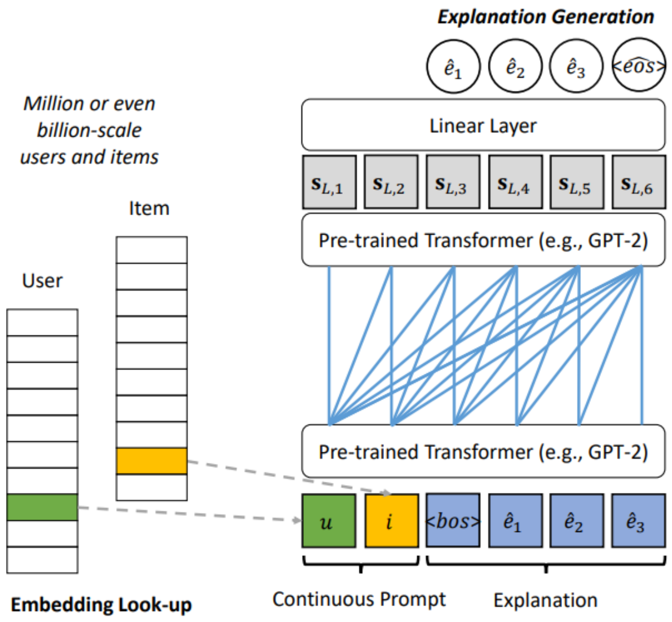


Fig. 4. Our proposed method PEPLER that treats user and item IDs as continuous prompt for explanation generation.

$$\mathcal{L}_C = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} \frac{1}{|E_{u,i}|} \sum_{t=1}^{|E_{u,i}|} - \log c_{2+t}^{e_t}$$

3 METHODOLOGY

3.4 EXPLANATION GENERATION

During the inference stage, our goal is to instruct the model to generate a word sequence E^*, which has the maximum log-likelihood, as explanation.

$$E^* = \arg\max_{E \in \hat{\mathcal{E}}} \sum_{t}^{|E|} \log c_{|prompt|+t}^{e_t} \tag{9}$$

where $\hat{\mathcal{E}}$ is the set of all generated word sequences, and $|prompt|$ denotes the prompt's length, i.e., 2 for $[u, i]$ and $|F_{u,i}|$ for $F_{u,i}$.

- Various methods to find the sequence E^* in text generation techniques, such as greedy decoding and beam search.
  - Adopt simple greedy encoding

Time step	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<eos>	0.1	0.2	0.2	0.6

Fig. 10.8.1 At each time step, greedy search selects the token with the highest conditional probability.

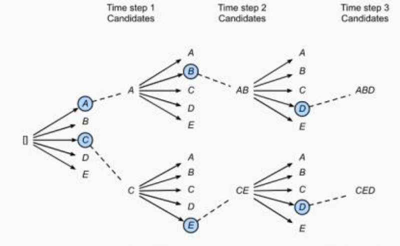


Fig. 10.8.3 The process of beam search (beam size: 2, maximum length of an output sequence: 3). The candidate output sequences are A, C, AB, CE, ABD, and CED.

3 METHODOLOGY

3.5 SEQUENTIAL TUNING STRATEGY –FOR CONTINUOUS PROMPT TUNING

In the case of discrete prompt learning, the prompts are features, which are of the same type as words that pre-trained language models were trained on. As a result, no additional model parameters are introduced, so we can simply optimize the following objective function:

$$\mathcal{J} = \min_{\Theta_{LM}} \mathcal{L}_D \tag{10}$$

where Θ_{LM} denotes all the trainable parameters in the pre-trained language model.

- In **continuous** prompt learning, we introduced additional prompt params, user and item embeddings
- Parameters needed to be updated include PLM params and prompt params
  - PLM params are already fully trained, while prompt params are randomly initialized

To tackle this problem, we propose a sequential tuning strategy. Specifically, we first freeze the language model parameters Θ_{LM}, and optimize the prompt parameters Θ_P with Eq. (8). Once Θ_P can no longer be updated, we fine-tune all the model parameters (i.e., Θ_{LM} and Θ_P) with Eq. (8) again. This two-step procedure can be demonstrated with the following formula:

$$\mathcal{J} = \min_{\Theta_P} \mathcal{L}_C \xrightarrow{\text{followed by}} \mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P\}} \mathcal{L}_C \tag{11}$$

3 METHODOLOGY

3.6 RECOMMENDATION AS REGULARIZATION –PEPLER

- Regularizing the learning of explanation generation via an additional **rating prediction** task
  - Motivation: **bridge** the aforementioned **gap** between pre-trained language models and continuous prompts
  - Intuition: rating captures the **relation** between user and item, to some extent
    - Recent studies find out that the two task of recommendation and an additional task (such as feature ranking, explanation ranking and review generation) could help the learning of each other
  - Adopt and test two **typical** recommendation models: MF, MLP

$$\hat{r}_{u,i} = \mathbf{u}^\top \mathbf{i}$$

$$\begin{cases} \mathbf{a}_0 = \sigma(\mathbf{W}_0[\mathbf{u}, \mathbf{i}] + \mathbf{b}_0) \\ \mathbf{a}_1 = \sigma(\mathbf{W}_1\mathbf{a}_0 + \mathbf{b}_1) \\ \ldots\ldots \\ \mathbf{a}_N = \sigma(\mathbf{W}_N\mathbf{a}_{N-1} + \mathbf{b}_N) \end{cases} \quad \text{and } \hat{r}_{u,i} = \mathbf{w}^\top \mathbf{a}_N + b$$

$$\mathcal{L}_R = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (r_{u,i} - \hat{r}_{u,i})^2$$

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_{REC}\}} (\mathcal{L}_C + \lambda\mathcal{L}_R)$$

3 METHODOLOGY

3.6 RECOMMENDATION AS REGULARIZATION –PEPLER

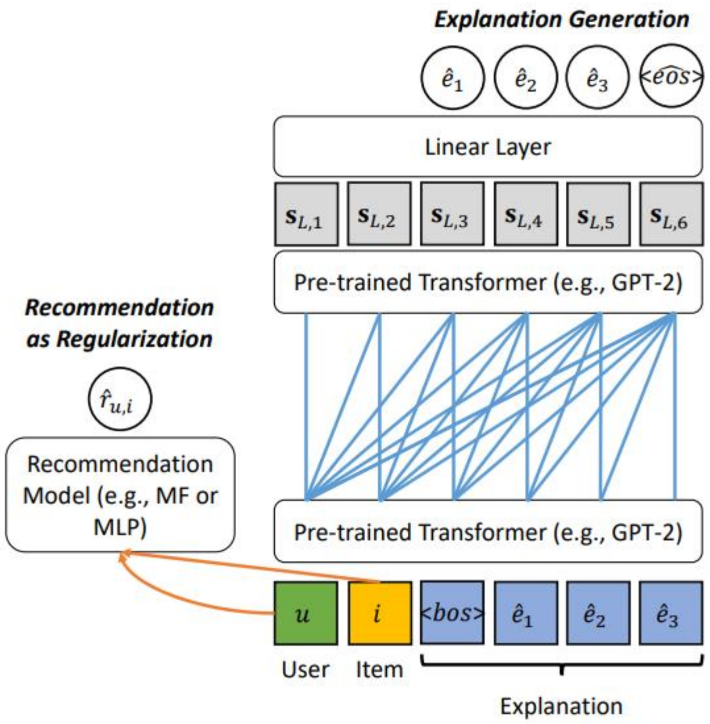


Fig. 5. Our proposed method PEPLER that regards the rating prediction task as a type of regularization for better learning of the explanation generation task.

4 EXPERIMENT SETUP

4.1 DATASETS

- Three publicly available explainable recommendation datasets
- Training, Validation, Testing = 8:1:1 for 5 times
- Training set holds at least one record for each user and each item
- Data format: user ID, item ID, rating from 1-5, explanation (extracted from user reviews, contains at least one item feature), feature

Table 4. Statistics of the three datasets.

	TripAdvisor	Amazon	Yelp
#users	9,765	7,506	27,147
#items	6,280	7,360	20,266
#records	320,023	441,783	1,293,247
#features	5,069	5,399	7,340
#records / user	32.77	58.86	47.64
#records / item	50.96	60.02	63.81
#words / explanation	13.01	14.14	12.32

4 EXPERIMENT SETUP

4.2 EVALUATION METRICS

- Measure perspectives
  - Text quality
    - BLEU (from machine translation, precision oriented), ROUGE (from text summarization, recall-oriented)
      - Not flawless, unable to detect the problem of identical sentences (same explanation for different user-item pairs
    - USR (Unique Sentence Ratio) unique sentences generated by a model (only holds one of the exactly matched explanations) divided by the total number of testing samples
    - Precision, Recall
  - Explainability
    - FMR (Feature Matching Ratio), FCR (Feature Coverage Ratio), DIV (Feature Diversity)

4 EXPERIMENT SETUP

4.2 EVALUATION METRICS

measure this, we adopt **USR** that computes the Unique Sentence Ratio of generated explanations [26]:

$$USR = \frac{|\mathcal{E}|}{N} \tag{16}$$

where $\mathcal{E}$ represents the set of unique sentences generated by a model, and N is the total number of testing samples. Note that, $\mathcal{E}$ only holds one of the exactly matched explanations.

FMR measures whether a generated explanation contains the feature in the ground-truth text. Formally, it is defined as follows:

$$FMR = \frac{1}{N} \sum_{u,i} \delta(f_{u,i} \in \hat{E}_{u,i}) \tag{17}$$

where $\hat{E}_{u,i}$ is the generated explanation for the user-item pair, $f_{u,i}$ is the feature in the ground-truth, and $\delta(x) = 1$ when x is true, or $\delta(x) = 0$ otherwise.

FCR is computed as the number of distinct features contained in all the generated explanations, divided by the total number of features in the whole dataset:

$$FCR = \frac{N_g}{|\mathcal{F}|} \tag{18}$$

where $\mathcal{F}$ is the collection of unique features in ground-truth explanations, and N_g denotes the amount of distinct features appeared in the generated explanations.

DIV measures the diversity of features between all generated explanations. The intuition is that explanations are expected to discuss different features in accordance with the given user-item pairs. Hence, it computes the intersection of features between any two generated explanations:

$$DIV = \frac{2}{N \times (N-1)} \sum_{u,u',i,i'} \left| \hat{\mathcal{F}}_{u,i} \cap \hat{\mathcal{F}}_{u',i'} \right| \tag{19}$$

4 EXPERIMENT SETUP

4.3 COMPARED METHODS

- Four state-of-the-art baselines: BERT, Transformer, GRU, LSTM
- Discrete prompt: do not directly use IDs but instead map IDs onto item feature
  - Aspect Conditional Masked Language Model (ACMLM) is a fine-tuned BERT
  - **PEPLER-D**
- Continuous prompt: leverage only user and item IDs to generate explanations
  - Neural Rating and Tips generation (NRT) can predict a rating and generate a tip simultaneously based on user and item IDs
  - Attribute-to-Sequence (Att2Seq) is a review generation approach with a two-layer LSTM
  - PErsonalized Transformer for Explainable Recommendation (PETER) is a small unpretrained Transformer particularly designed for explanation generation
  - **PEPLER**

4 EXPERIMENT SETUP

4.4 IMPLEMENTATION DETAILS

- Results are averaged on the 5 data splits
- Optimize PEPLER and PEPLER-D with AdamW
- Inspect validation loss, if not decrease for 5 epochs, then stop training and load the best model for prediction
- Regularization coefficient from $[10^{-3}, 10^{-2}, \ldots, 10^{3}]$

5 RESULTS AND ANALYSIS

5.1 QUANTITATIVE ANALYSIS ON EXPLANATIONS

Table 5. Performance comparison of explanation generation methods in terms of Explainability and Text Quality on three datasets. The methods are divided into two groups according to whether IDs are directly used or not. B1 and B4 stand for BLEU-1 and BLEU-4. R1-P, R1-R, R1-F, R2-P, R2-R and R2-F denote Precision, Recall and F1 of ROUGE-1 and ROUGE-2. BLEU and ROUGE are percentage values (% symbol omitted for table clarity), while the others are absolute values. The best performing values are boldfaced, and ** and * indicate the statistical significance over the second best baseline respectively for $p < 0.01$ and $p < 0.05$ via Student's t-test.

	Explainability				Text Quality							
	FMR↑	FCR↑	DIV↓	USR↑	B1↑	B4↑	R1-P↑	R1-R↑	R1-F↑	R2-P↑	R2-R↑	R2-F↑
	Yelp											
ACMLM	**0.05**	**0.31**	**0.95**	**0.95**	7.01	0.24	7.89	7.54	6.82	0.44	0.48	0.39
PEPLER-D	**0.05**	0.24	1.53	0.13	**9.17****	**0.40****	**15.67****	**10.47****	**11.73****	**1.09****	**0.78****	**0.83****
NRT	0.06	0.12	1.67	0.20	10.92	0.60	16.73	11.91	12.89	1.63	1.21	1.26
Att2Seq	0.05	0.05	2.25	0.05	10.25	0.54	17.13	11.44	12.72	1.49	1.13	1.16
PETER	**0.08**	0.15	1.62	0.15	10.74	0.63	16.18	11.90	12.63	1.60	1.32	1.28
PEPLER	**0.08****	**0.30****	**1.52**	**0.35****	**11.23**	**0.73****	**17.51**	**12.55***	**13.53****	**1.86***	**1.42**	**1.46****
	Amazon											
ACMLM	**0.10**	**0.31**	2.07	**0.96**	9.52	0.22	11.65	10.39	9.69	0.71	0.81	0.64
PEPLER-D	0.08	0.19	**1.85***	0.15	**10.94****	**0.49****	**16.31****	**11.80****	**12.80****	**1.43****	**1.13****	**1.16****
NRT	0.10	0.04	2.71	0.09	12.06	0.69	17.17	13.15	13.83	1.94	1.68	1.64
Att2Seq	0.09	0.04	2.64	0.05	12.07	0.73	18.35	12.86	14.14	2.01	1.56	1.61
PETER	0.09	0.09	2.16	0.20	11.75	0.89	16.51	13.10	13.55	1.96	1.76	1.68
PEPLER	**0.11**	**0.27****	**2.06**	**0.38****	**13.19***	**1.05****	**18.51**	**14.16**	**14.87**	**2.36***	**1.88**	**1.91****
	TripAdvisor											
ACMLM	**0.07**	**0.41**	**0.78**	**0.94**	3.45	0.02	4.86	3.82	3.72	0.18	0.20	0.16
PEPLER-D	0.05	0.22	2.69	0.08	**14.61****	**0.87****	**18.07****	**14.83****	**15.32****	**1.76****	**1.66****	**1.58****
NRT	0.05	0.02	6.07	0.00	13.76	0.80	19.01	14.57	15.58	2.10	1.59	1.68
Att2Seq	0.06	0.05	4.74	0.02	15.20	0.96	18.74	**16.42**	**16.38**	2.42	**2.32**	**2.19**
PETER	**0.07**	0.09	3.62	0.05	15.13	1.00	18.30	16.15	16.00	2.24	2.23	2.06
PEPLER	**0.07***	**0.21****	**2.71****	**0.24****	**15.49**	**1.09**	**19.48**	15.67	16.24	**2.48**	2.21	2.16

5 RESULTS AND ANALYSIS

5.1 QUANTITATIVE ANALYSIS ON EXPLANATIONS

- Discrete prompt
  - PEPLER-D consistently and significantly beats ACMLM
    - PEPLER-D's effectiveness in generating high-quality sentences that are semantically close to the ground-truth text
    - ACMLM (a fine-tuned BERT) falls behind by a large gap, because the its generation is achieved by predicting masked tokens, which is quite different from conventional auto-regressive generation.
    - This may explain why ACMLM produces diverse sentences (high USR) and features (low DIV), which, however, is less meaningful when text quality cannot be guaranteed.

Table 5. Performance comparison of explanation generation methods in terms of Explainability and Text Quality on three datasets. The methods are divided into two groups according to whether IDs are directly used or not. B1 and B4 stand for BLEU-1 and BLEU-4. R1-P, R1-R, R1-F, R2-P, R2-R and R2-F denote Precision, Recall and F1 of ROUGE-1 and ROUGE-2. BLEU and ROUGE are percentage values (% symbol omitted for table clarity), while the others are absolute values. The best performing values are boldfaced, and ** and * indicate the statistical significance over the second best baseline respectively for $p < 0.01$ and $p < 0.05$ via Student's t-test.

	Explainability				Text Quality							
	FMR↑	FCR↑	DIV↓	USR↑	B1↑	B4↑	R1-P↑	R1-R↑	R1-F↑	R2-P↑	R2-R↑	R2-F↑
	Yelp											
ACMLM	**0.05**	**0.31**	**0.95**	**0.95**	7.01	0.24	7.89	7.54	6.82	0.44	0.48	0.39
PEPLER-D	**0.05**	0.24	1.53	0.13	**9.17****	**0.40****	**15.67****	**10.47****	**11.73****	**1.09****	**0.78****	**0.83****
NRT	0.06	0.12	1.67	0.20	10.92	0.60	16.73	11.91	12.89	1.63	1.21	1.26
Att2Seq	0.05	0.05	2.25	0.05	10.25	0.54	17.13	11.44	12.72	1.49	1.13	1.16
PETER	**0.08**	0.15	1.62	0.15	10.74	0.63	16.18	11.90	12.63	1.60	1.32	1.28
PEPLER	**0.08****	**0.30****	**1.52**	**0.35****	**11.23**	**0.73****	**17.51**	**12.55***	**13.53****	**1.86***	**1.42**	**1.46****
	Amazon											
ACMLM	**0.10**	**0.31**	2.07	**0.96**	9.52	0.22	11.65	10.39	9.69	0.71	0.81	0.64
PEPLER-D	0.08	0.19	**1.85***	0.15	**10.94****	**0.49****	**16.31****	**11.80****	**12.80****	**1.43****	**1.13****	**1.16****
NRT	0.10	0.04	2.71	0.09	12.06	0.69	17.17	13.15	13.83	1.94	1.68	1.64
Att2Seq	0.09	0.04	2.64	0.05	12.07	0.73	18.35	12.86	14.14	2.01	1.56	1.61
PETER	0.09	0.09	2.16	0.20	11.75	0.89	16.51	13.10	13.55	1.96	1.76	1.68
PEPLER	**0.11**	**0.27****	**2.06**	**0.38****	**13.19***	**1.05****	**18.51**	**14.16**	**14.87**	**2.36***	**1.88**	**1.91****
	TripAdvisor											
ACMLM	**0.07**	**0.41**	**0.78**	**0.94**	3.45	0.02	4.86	3.82	3.72	0.18	0.20	0.16
PEPLER-D	0.05	0.22	2.69	0.08	**14.61****	**0.87****	**18.07****	**14.83****	**15.32****	**1.76****	**1.66****	**1.58****
NRT	0.05	0.02	6.07	0.00	13.76	0.80	19.01	14.57	15.58	2.10	1.59	1.68
Att2Seq	0.06	0.05	4.74	0.02	15.20	0.96	18.74	**16.42**	**16.38**	2.42	**2.32**	**2.19**
PETER	**0.07**	0.09	3.62	0.05	15.13	1.00	18.30	16.15	16.00	2.24	2.23	2.06
PEPLER	**0.07***	**0.21****	**2.71****	**0.24****	**15.49**	**1.09**	**19.48**	15.67	16.24	**2.48**	2.21	2.16

5 RESULTS AND ANALYSIS

5.1 QUANTITATIVE ANALYSIS ON EXPLANATIONS

- Continuous prompt
  - Text quality much better than discrete approaches
  - RNN-based NRT and Att2Seq may suffer from long-term dependency problem, not comparable to PETER and PEPLER
  - PETER is a small unpretrained Transformer, without rich linguistic knowledge from pre-training
  - Sequential tuning could effectively make use of such knowledge for generating better explanations

Table 5. Performance comparison of explanation generation methods in terms of Explainability and Text Quality on three datasets. The methods are divided into two groups according to whether IDs are directly used or not. B1 and B4 stand for BLEU-1 and BLEU-4. R1-P, R1-R, R1-F, R2-P, R2-R and R2-F denote Precision, Recall and F1 of ROUGE-1 and ROUGE-2. BLEU and ROUGE are percentage values (% symbol omitted for table clarity), while the others are absolute values. The best performing values are boldfaced, and ** and * indicate the statistical significance over the second best baseline respectively for $p < 0.01$ and $p < 0.05$ via Student's t-test.

	Explainability				Text Quality							
	FMR↑	FCR↑	DIV↓	USR↑	B1↑	B4↑	R1-P↑	R1-R↑	R1-F↑	R2-P↑	R2-R↑	R2-F↑
	Yelp											
ACMLM	**0.05**	**0.31**	**0.95**	**0.95**	7.01	0.24	7.89	7.54	6.82	0.44	0.48	0.39
PEPLER-D	**0.05**	0.24	1.53	0.13	**9.17****	**0.40****	**15.67****	**10.47****	**11.73****	**1.09****	**0.78****	**0.83****
NRT	0.06	0.12	1.67	0.20	10.92	0.60	16.73	11.91	12.89	1.63	1.21	1.26
Att2Seq	0.05	0.05	2.25	0.05	10.25	0.54	17.13	11.44	12.72	1.49	1.13	1.16
PETER	**0.08**	0.15	1.62	0.15	10.74	0.63	16.18	11.90	12.63	1.60	1.32	1.28
PEPLER	**0.08****	**0.30****	**1.52**	**0.35****	**11.23**	**0.73****	**17.51**	**12.55***	**13.53****	**1.86***	**1.42**	**1.46****
	Amazon											
ACMLM	**0.10**	**0.31**	2.07	**0.96**	9.52	0.22	11.65	10.39	9.69	0.71	0.81	0.64
PEPLER-D	0.08	0.19	**1.85***	0.15	**10.94****	**0.49****	**16.31****	**11.80****	**12.80****	**1.43****	**1.13****	**1.16****
NRT	0.10	0.04	2.71	0.09	12.06	0.69	17.17	13.15	13.83	1.94	1.68	1.64
Att2Seq	0.09	0.04	2.64	0.05	12.07	0.73	18.35	12.86	14.14	2.01	1.56	1.61
PETER	0.09	0.09	2.16	0.20	11.75	0.89	16.51	13.10	13.55	1.96	1.76	1.68
PEPLER	**0.11**	**0.27****	**2.06**	**0.38****	**13.19***	**1.05****	**18.51**	**14.16**	**14.87**	**2.36***	**1.88**	**1.91****
	TripAdvisor											
ACMLM	**0.07**	**0.41**	**0.78**	**0.94**	3.45	0.02	4.86	3.82	3.72	0.18	0.20	0.16
PEPLER-D	0.05	0.22	2.69	0.08	**14.61****	**0.87****	**18.07****	**14.83****	**15.32****	**1.76****	**1.66****	**1.58****
NRT	0.05	0.02	6.07	0.00	13.76	0.80	19.01	14.57	15.58	2.10	1.59	1.68
Att2Seq	0.06	0.05	4.74	0.02	15.20	0.96	18.74	**16.42**	**16.38**	2.42	**2.32**	**2.19**
PETER	**0.07**	0.09	3.62	0.05	15.13	1.00	18.30	16.15	16.00	2.24	2.23	2.06
PEPLER	**0.07***	**0.21****	**2.71****	**0.24****	**15.49**	**1.09**	**19.48**	15.67	16.24	**2.48**	2.21	2.16

5 RESULTS AND ANALYSIS

5.1 QUANTITATIVE ANALYSIS ON EXPLANATIONS

Table 5. Performance comparison of explanation generation methods in terms of Explainability and Text Quality on three datasets. The methods are divided into two groups according to whether IDs are directly used or not. B1 and B4 stand for BLEU-1 and BLEU-4. R1-P, R1-R, R1-F, R2-P, R2-R and R2-F denote Precision, Recall and F1 of ROUGE-1 and ROUGE-2. BLEU and ROUGE are percentage values (% symbol omitted for table clarity), while the others are absolute values. The best performing values are boldfaced, and ** and * indicate the statistical significance over the second best baseline respectively for $p < 0.01$ and $p < 0.05$ via Student's t-test.

	Explainability				Text Quality							
	FMR↑	FCR↑	DIV↓	USR↑	B1↑	B4↑	R1-P↑	R1-R↑	R1-F↑	R2-P↑	R2-R↑	R2-F↑
	Yelp											
ACMLM	**0.05**	**0.31**	**0.95**	**0.95**	7.01	0.24	7.89	7.54	6.82	0.44	0.48	0.39
PEPLER-D	**0.05**	0.24	1.53	0.13	**9.17****	**0.40****	**15.67****	**10.47****	**11.73****	**1.09****	**0.78****	**0.83****
NRT	0.06	0.12	1.67	0.20	10.92	0.60	16.73	11.91	12.89	1.63	1.21	1.26
Att2Seq	0.05	0.05	2.25	0.05	10.25	0.54	17.13	11.44	12.72	1.49	1.13	1.16
PETER	**0.08**	0.15	1.62	0.15	10.74	0.63	16.18	11.90	12.63	1.60	1.32	1.28
PEPLER	**0.08****	**0.30****	**1.52**	**0.35****	**11.23**	**0.73****	**17.51**	**12.55***	**13.53****	**1.86***	**1.42**	**1.46****
	Amazon											
ACMLM	**0.10**	**0.31**	2.07	**0.96**	9.52	0.22	11.65	10.39	9.69	0.71	0.81	0.64
PEPLER-D	0.08	0.19	**1.85***	0.15	**10.94****	**0.49****	**16.31****	**11.80****	**12.80****	**1.43****	**1.13****	**1.16****
NRT	0.10	0.04	2.71	0.09	12.06	0.69	17.17	13.15	13.83	1.94	1.68	1.64
Att2Seq	0.09	0.04	2.64	0.05	12.07	0.73	18.35	12.86	14.14	2.01	1.56	1.61
PETER	0.09	0.09	2.16	0.20	11.75	0.89	16.51	13.10	13.55	1.96	1.76	1.68
PEPLER	**0.11**	**0.27****	**2.06**	**0.38****	**13.19***	**1.05****	**18.51**	**14.16**	**14.87**	**2.36***	**1.88**	**1.91****
	TripAdvisor											
ACMLM	**0.07**	**0.41**	**0.78**	**0.94**	3.45	0.02	4.86	3.82	3.72	0.18	0.20	0.16
PEPLER-D	0.05	0.22	2.69	0.08	**14.61****	**0.87****	**18.07****	**14.83****	**15.32****	**1.76****	**1.66****	**1.58****
NRT	0.05	0.02	6.07	0.00	13.76	0.80	19.01	14.57	15.58	2.10	1.59	1.68
Att2Seq	0.06	0.05	4.74	0.02	15.20	0.96	18.74	**16.42**	**16.38**	2.42	**2.32**	**2.19**
PETER	**0.07**	0.09	3.62	0.05	15.13	1.00	18.30	16.15	16.00	2.24	2.23	2.06
PEPLER	**0.07***	**0.21****	**2.71****	**0.24****	**15.49**	**1.09**	**19.48**	15.67	16.24	**2.48**	2.21	2.16

- Continuous prompt
  - Some special cases on the TripAdvisor dataset, where Att2Seq obtains the largest ROUGE scores
    - Fix its generation issue? Making it a competitive baseline?
    - Small datasets may cause large PEPLER to underfit (not in real world , not in large datasets like Yelp)

5 RESULTS AND ANALYSIS

5.2 EFFECT OF SEQUENTIAL TUNING

- To validate the superiority of our proposed Sequential Tuning strategy, we compare it with its two composite training strategies: Fixed-LM Prompt Tuning and Prompt+LM Fine-tuning

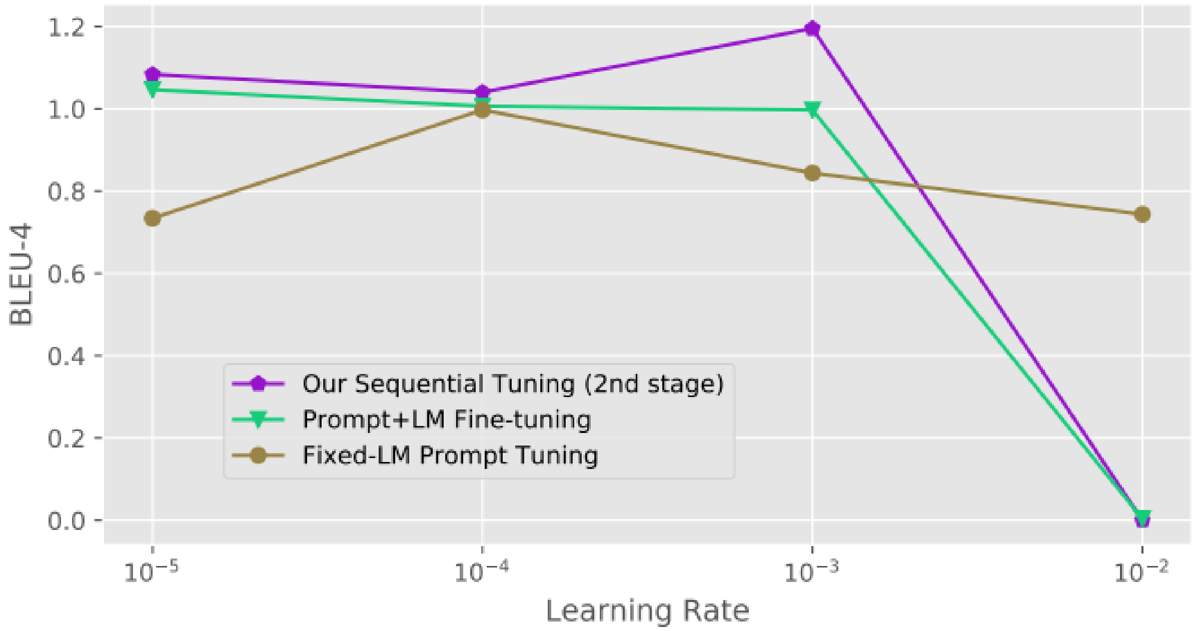


Fig. 6. A comparison of three tuning strategies for prompt learning in terms of BLEU-4 with varying learning rates on the TripAdvisor dataset.

5 RESULTS AND ANALYSIS

5.2 EFFECT OF SEQUENTIAL TUNING

- **PEPLER achieved highest** BLEU-4 score, manifesting the advantage in bridging the gap between the randomly initialized continuous prompts and the pre-trained language model.
- The pattern of our Sequential Tuning and that of Prompt+LM Fine-tuning (green) is quite **similar**, with all the params tuned by both.
- Sequential tuning outperforms Prompt+LM Fine-tuning since the former's prompts are **already trained**.
- Large learning rate may cause significant changes of parameters in the pre-train model, leading to severe **performance degradation**.
- Fixed-LM Prompt Tuning is quite **agnostic** to learning rate changes but not outperforming the other two strategies, because the model is **frozen** and only prompts can be tuned, and therefore could not be well **adjusted** to the target explanation task.

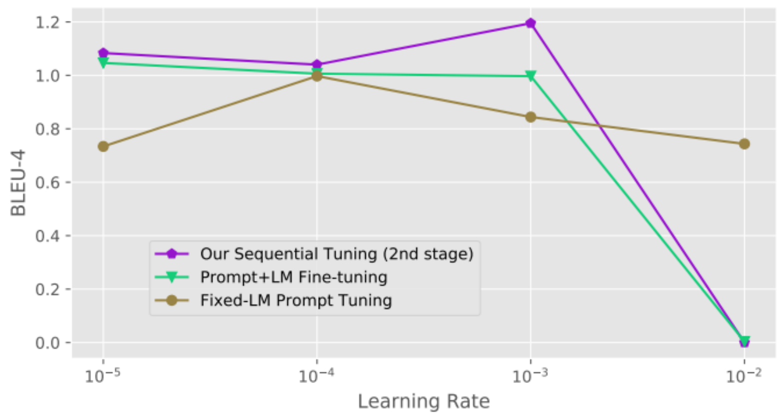


Fig. 6. A comparison of three tuning strategies for prompt learning in terms of BLEU-4 with varying learning rates on the TripAdvisor dataset.

5 RESULTS AND ANALYSIS

5.2 EFFECT OF RECOMMENDATION AS REGULARIZATION

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_{REC}\}} (\mathcal{L}_C + \lambda\mathcal{L}_R)$$

- The default PEPLER employs sequential tuning, while the other two methods utilize recommendation as regularization with respectively MF (denoted as PEPLER+MF) and MLP (represented by PEPLER+MLP)

Table 6. Comparison of our proposed two strategies for continuous prompt learning on the TripAdvisor dataset. PEPLER employs the default sequential tuning, while the other two methods use recommendation as regularization with MF and MLP respectively. "Rec Para." stands for additional parameters for recommendation. Arrows ↑ and ↓ respectively denote the performance increase and decrease compared with PEPLER.

	Explainability				Text Quality		Training		
	FMR	FCR	DIV	USR	BLEU-1	BLEU-4	Epoch	Rating	Rec Para.
PEPLER	0.07	0.21	2.71	0.24	15.49	1.09	17	No	No
PEPLER+MF	0.08↑	0.26↑	3.00↓	0.29↑	16.67↑	1.27↑	10	Yes	No
PEPLER+MLP	0.07	0.09↓	3.36↓	0.07↓	16.05↑	1.09	8	Yes	Yes

5 RESULTS AND ANALYSIS

5.2 EFFECT OF RECOMMENDATION AS REGULARIZATION

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_{REC}\}} (\mathcal{L}_C + \lambda \mathcal{L}_R)$$

- Compared with PEPLER, PEPLER+MF not only improves the **text quality** but also **explainability**
- PEPLER+MLP maintains **comparable text quality** to PEPLER, but cannot keep up the **explainability**, e.g., the decreasing in FCR and USR
  - MLP has linear layer parameters while MF does not. These parameters might help to predict ratings but adversely affect the learning of the explanation task
- Since the recommendation task requires extra rating data for training, which may not be always available in other natural language generation tasks (e.g., dialogue systems), we set sequential tuning as the **default** training strategy for PEPLER.
- This strategy needs more training epochs, because it has two stages. Depending on the specific application, one may consider PEPLER+MF

Table 6. Comparison of our proposed two strategies for continuous prompt learning on the TripAdvisor dataset. PEPLER employs the default sequential tuning, while the other two methods use recommendation as regularization with MF and MLP respectively. "Rec Para." stands for additional parameters for recommendation. Arrows ↑ and ↓ respectively denote the performance increase and decrease compared with PEPLER.

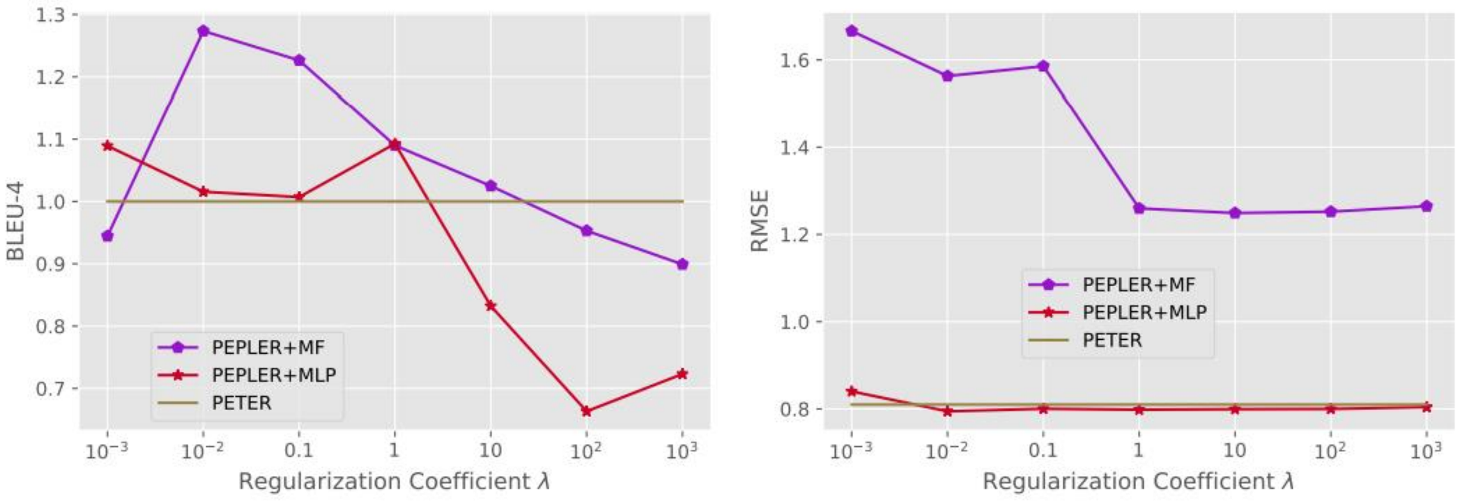
	Explainability				Text Quality		Training		
	FMR	FCR	DIV	USR	BLEU-1	BLEU-4	Epoch	Rating	Rec Para.
PEPLER	0.07	0.21	2.71	0.24	15.49	1.09	17	No	No
PEPLER+MF	0.08↑	0.26↑	3.00↓	0.29↑	16.67↑	1.27↑	10	Yes	No
PEPLER+MLP	0.07	0.09↓	3.36↓	0.07↓	16.05↑	1.09	8	Yes	Yes

5 RESULTS AND ANALYSIS

5.2 EFFECT OF RECOMMENDATION AS REGULARIZATION

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_{REC}\}} (\mathcal{L}_C + \lambda \mathcal{L}_R)$$

- Investigate how PEPLER+MF and PEPLER+MLP react to varying λ, the regularization coefficient on the recommendation task



(a) BLEU-4 with varying λ (b) RMSE with varying λ

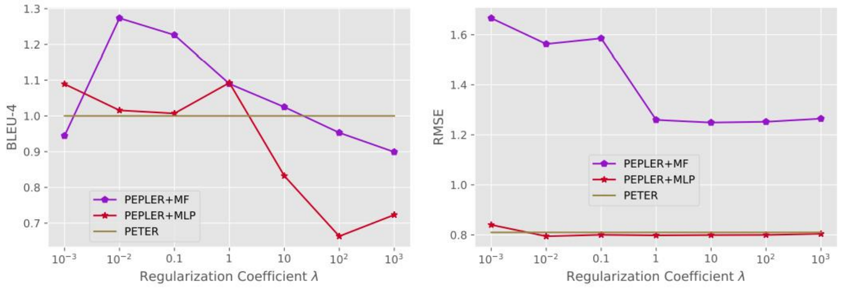
Fig. 7. The effect of regularization coefficient λ on the recommendation task with MF or MLP for PEPLER on the TripAdvisor dataset. For better comparison, the results of PETER are shown.

5 RESULTS AND ANALYSIS

5.3 EFFECT OF RECOMMENDATION AS REGULARIZATION

$$\mathcal{J} = \min_{\Theta=\{\Theta_{LM},\Theta_P,\Theta_{REC}\}} (\mathcal{L}_C + \lambda \mathcal{L}_R)$$

- Investigate how PEPLER+MF and PEPLER+MLP react to varying λ, the regularization coefficient on the recommendation task
- **Trade-off** between explanation performance and recommendation performance
  - Model with small λ (no greater than 1) can reach an optimal point for explanation performance but explodes for recommendation performance
  - Support the design of leveraging recommendation task to help the learning of explanation generation
- PEPLER+MLP outperforms PEPLER+MF by a large margin. Because MLP can model more complex relations between users and items than MF.



(a) BLEU-4 with varying λ (b) RMSE with varying λ

Fig. 7. The effect of regularization coefficient λ on the recommendation task with MF or MLP for PEPLER on the TripAdvisor dataset. For better comparison, the results of PETER are shown.

5 RESULTS AND ANALYSIS

5.4 QUALITATIVE CASE STUDY ON EXPLANATIONS

- Two examples generated by all the methods for hotel recommendations on the TripAdvisor dataset.

Table 7. Explanations on two different cases as generated by different methods on the TripAdvisor dataset. The boldfaced words in the ground-truth are the key features. Matched features in the generated explanations are also boldfaced.

Ground-truth	the swimming **pool** is fantastic
ACMLM	swimming **pool** swimming **pools pool** strip beach area
NRT	the hotel is located in a great location
Att2Seq	the hotel is located in the heart of the city and the main shopping area is also within walking distance
PETER	the hotel is located in the heart of the city and the harbour
PEPLER-D	the room was very nice and the bed was very comfortable
PEPLER	the **pool** is amazing and the **pool** is very relaxing
Ground-truth	this is one of the finest **hotels** in all of Europe
ACMLM	swimming pool area pool ja ##cu ##zzi pool city area gym building pool area spa gym pool area
NRT	the **hotel** is located in a great location
Att2Seq	the **hotel** is located in the heart of the city and the main shopping area is also within walking distance
PETER	the **hotel** is in a great location
PEPLER-D	the **hotel** is a short walk from the old town
PEPLER	the **hotel** is located in the heart of the city and is very well maintained

5 RESULTS AND ANALYSIS

5.4 QUALITATIVE CASE STUDY ON EXPLANATIONS

- ACMLM: unreadable (BLEU and ROUGE, text quality)
- Att2Seq tends to generate **long** explanations, which may explain why it obtains good performance regarding ROUGE on the TripAdvisor dataset (see Table 5), because ROUGE is a recall-oriented metric and favors long sentences.
- PETER, PEPLER-D, PEPLER are quite good owing to **powerful language modeling capability of Transformer**
- **Rich expression** of PEPLER's explanation, thanks to the linguistic language contained in the **pre-trained** model

Table 7. Explanations on two different cases as generated by different methods on the TripAdvisor dataset. The boldfaced words in the ground-truth are the key features. Matched features in the generated explanations are also boldfaced.

Ground-truth	the swimming **pool** is fantastic
ACMLM	swimming **pool** swimming **pools** **pool** strip beach area
NRT	the hotel is located in a great location
Att2Seq	the hotel is located in the heart of the city and the main shopping area is also within walking distance
PETER	the hotel is located in the heart of the city and the harbour
PEPLER-D	the room was very nice and the bed was very comfortable
PEPLER	the **pool** is amazing and the **pool** is very relaxing
Ground-truth	this is one of the finest **hotels** in all of Europe
ACMLM	swimming pool area pool ja ##cu ##zzi pool city area gym building pool area spa gym pool area
NRT	the **hotel** is located in a great location
Att2Seq	the **hotel** is located in the heart of the city and the main shopping area is also within walking distance
PETER	the **hotel** is in a great location
PEPLER-D	the **hotel** is a short walk from the old town
PEPLER	the **hotel** is located in the heart of the city and is very well maintained

6 CONCLUSION & FUTURE WORK

6.1 CONCLUSION

- Propose **two prompt learning approaches** (discrete/continuous prompt) for recommendation explanation generation
- **Two learning strategies** (Sequential tuning and recommendation as regularization) to bridge the gap between continuous prompts and pre-trained models
- Extensive experiments demonstrate the **effectiveness** of the proposed approaches in generating high-quality explanations as measured by text quality and explainability metrics

6 CONCLUSION & FUTURE WORK

6.2 FUTURE WORK

- **Debias** in prompt learning (discrimination against the minority)
- Apply the proposed approaches to **other applications** of personalized natural language generation, such as personalized question answering systems and personalized conversational agents
- **Incorporate item images** into pre-trained models to generate visual explanations for recommendations
- Adapt pre-trained models to **cross-lingual explanation** generation, since international platforms

7 COMMENTS & TAKEAWAY

7.1 COMMENTS

- Pros
  - Nice try of prompt learning on recommendation scenario. Explanation generation fits the PLM training objectives. **FIRST** to introduce prompt learning to the community of recommender systems
  - **Comprehensive try** of prompt learning (discrete and continuous prompt)
- Cons
  - Only migrates prompt learning to recommendation. **Lack of substantial innovations**
  - Sequential tuning, though effective, still needs to fine-tune the PLM, which **breaks the subtle elegance of prompt learning**
  - Recommendation as regularization, despite its effectiveness, seems a little redundant and **unnecessary**

7 COMMENTS & TAKEAWAY

7.2 TAKEAWAY

- Prompt learning is a new effective paradigm in Natural Language Processing, which avoids huge efforts to fine-tune PLMs and takes the advantage of PLMs' rich linguistics knowledge. Prompt engineering and hallucination are potential risks of prompt learning
- Prompt learning can be applied to recommendation explanations, achieving promising performance
- **FIND THE POTENTIAL EMERGING RESEARCH DIRECTIONS AND TAKE THE LEAD ASAP!**

REFENRENCES (NOT ALL)

- https://arxiv.org/abs/2202.07371
- https://arxiv.org/pdf/2107.13586.pdf
- https://www.jiqizhixin.com/articles/2021-08-03-7
- https://www.reinfer.io/blog/prompt-based-learning
- https://developers.reinfer.io/blog/2022/05/04/prompting
- https://arxiv.org/pdf/2202.03629.pdf
- https://github.com/thunlp/PromptPapers
- https://jalammar.github.io/illustrated-transformer/
- https://arxiv.org/pdf/1810.04805.pdf

PERSONALIZED PROMPT LEARNING FOR EXPLAINABLE RECOMMENDATION

STUDY GROUP

THUIR

任一 renyi1006@gmail.com

2022/09/15

MINI HALL, FIT 1-506, TSINGHUA UNIVERSITY